Theorem Proving Modulo Revised Version

Gilles Dowek, Thérèse Hardin, Claude Kirchner

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Theorem Proving Modulo Revised Version

Gilles Dowek, Thérèse Hardin, Claude Kirchner

No 4861

June 2003

THÈME 2

ISSN 0249-6399 ISRN INRIA/RR--4861--FR+ENG

Theorem Proving Modulo
Revised Version

Gilles Dowek[*], **Thérèse Hardin**[†], **Claude Kirchner**[‡]

Thème 2 — Génie logiciel
et calcul symbolique
Projets Logical, Moscova, Protheo

Rapport de recherche n° 4861 — June 2003 — 54 pages

Abstract: *Deduction modulo* is a way to remove computational arguments from proofs by reasoning modulo a congruence on propositions. Such a technique, issued from automated theorem proving, is of general interest because it permits to separate computations and deductions in a clean way. The first contribution of this paper is to define a *sequent calculus modulo* that gives a proof theoretic account of the combination of computations and deductions. The congruence on propositions is handled via rewrite rules and equational axioms. Rewrite rules apply to terms but also directly to atomic propositions.

The second contribution is to give a complete proof search method, called *Extended Narrowing and Resolution* (ENAR), for theorem proving modulo such congruences. The completeness of this method is proved with respect to provability in sequent calculus modulo.

An important application is that higher-order logic can be presented as a theory in deduction modulo. Applying the *Extended Narrowing and Resolution* method to this presentation of higher-order logic subsumes full higher-order resolution.

Key-words: Automated theorem proving, rewriting, resolution, narrowing, higher-order logic, cut elimination, deduction modulo, sequent calculus modulo, skolemization.

(Résumé : tsvp)

[*] INRIA-Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France, Gilles.Dowek@inria.fr
[†] LIP6 and INRIA-Rocquencourt, Université Paris 6, 4, Place Jussieu, 75252 Paris Cedex 05, France, Therese.Hardin@lip6.fr
[‡] LORIA and INRIA, 615, rue du Jardin Botanique, B.P. 101, 54602 Villers-lès-Nancy Cedex, France, Claude.Kirchner@loria.fr

Unité de recherche INRIA Rocquencourt
Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
Téléphone : 01 39 63 55 11 - International : +33 1 39 63 55 11
Télécopie : (33) 01 39 63 53 30 - International : +33 1 39 63 53 30

Preuve de Théorème Modulo
Version Révisée

Résumé : La "déduction modulo" est un moyen de supprimer les phases de calcul apparaissant dans les démonstrations en raisonnant modulo une congruence décrivant les calculs effectués sur les termes et les propositions. Cette technique, issue de la démonstration automatique, est d'une portée plus générale car elle permet d'identifier et combiner les phases de raisonnement et les phases de calcul. La première contribution de ce papier est la définition d'un *calcul des séquents modulo* permettant la combinaison du calcul et de la déduction. La congruence sur les propositions est définie par un ensemble de règles de réécriture et d'axiomes équationnels. Les règles de réécriture s'appliquant ici non seulement sur les termes mais aussi sur les propositions.

La seconde contribution de ce travail est de donner une méthode complète de démonstration automatique, appellée "Surréduction et résolution étendues" (ENAR), modulo de telles congruences. La complétude de cette méthode est établie en se basant sur la prouvabilité dans le calcul des séquents modulo.

Une application importante concerne la logique d'ordre supérieur qui peut être représentée comme une théorie en déduction modulo. L'application de la méthode de recherche de preuve ENAR à cette présentation de la logique d'ordre supérieur permet alors d'exprimer la résolution d'ordre supérieur.

Mots-clé : Démonstration automatique, réécriture, résolution, surréduction, logique d'ordre supérieur, élimination des coupures, déduction modulo, calcul des séquents modulo, skolémisation.

Introduction

The use of rewriting in automated theorem proving

If let loose, usual automated theorem proving methods such as resolution or tableaux can be very inefficient, spending time in trivial deductions. For instance, trying to prove the proposition

$$(a + b) + ((c + d) + e) = a + ((b + c) + (d + e))$$

with the associativity axiom

$$\forall x\ \forall y\ \forall z\ ((x + y) + z = x + (y + z))$$

and the identity axiom

$$\forall x\ x = x$$

a naive program may endlessly rearrange brackets before finding the right derivation. In contrast, humans would think that this problem requires no *deduction*, but rather just a *computation*. They would apply a deterministic and terminating algorithm to check that these two terms are the same modulo associativity. Hence the distinction between computation and deduction seems to be of prime interest when solving mathematical problems. Computations require just blind execution while deductions need non-deterministic search.

Computation can be defined using a decidable congruence on propositions: in the example above this congruence identifies propositions equivalent modulo rearrangement of brackets. Such decidable congruences can often be defined by confluent and terminating systems of rewrite rules. In some cases, equational axioms can also be considered. Deduction can be defined using inference rules and axioms, as it is in sequent calculus, natural deduction, Hilbert's systems, ... Deduction rules and computation rules should not be confused as they model very different activities.

Automated theorem proving methods benefit from such a distinction between deduction and computation. For instance, the associativity axiom can be replaced by the rewrite rule

$$(x + y) + z \to x + (y + z)$$

Then, the proposition

$$\neg((a + b) + ((c + d) + e) = a + ((b + c) + (d + e)))$$

rewrites to

$$\neg(a + (b + (c + (d + e))) = a + (b + (c + (d + e))))$$

that gives directly a refutation with the clause

$$x = x$$

Each step of the rewriting derivation can be simulated in resolution with the associativity axiom and the axioms of equality. But, unlike the search for a refutation derivation, rewriting

can be used in a deterministic way, provided the rewrite system is confluent. Hence, the rewriting derivation does the job of *all* the mentioned refutation derivations at once.

In the example above, computation rules apply to the terms of the language. As already remarked by S.J. Lee, D. Plaisted and R. Potter [Plaisted and Potter, 1991, Lee and Plaisted, 1994] such rules can also apply to the propositions of the language. Let us for example consider the axioms $P \Leftrightarrow (Q \vee R)$, $Q \Leftrightarrow S$, $R \Leftrightarrow S$, P, $\neg S$. There are many ways, equivalent up to permutation of proof steps, to refute by resolution its clausal variant

$$\neg P, Q, R \quad \neg Q, P \quad \neg R, P \quad \neg Q, S \quad \neg S, Q \quad \neg R, S \quad \neg S, R \quad P \quad \neg S$$

But the proposition $P \Leftrightarrow (Q \vee R)$ can be considered as a definition of P and similarly the propositions $Q \Leftrightarrow S$ and $R \Leftrightarrow S$ as definitions of Q and R. This permits to suppress these axioms and to work modulo the congruence defined by the rewrite rules:

$$\begin{array}{lcl} P & \to & Q \vee R \\ Q & \to & S \\ R & \to & S \end{array}$$

that form a confluent and terminating rewrite system on propositions. So only two propositions remain: P and $\neg S$. The proposition P unfolds (i.e. rewrites) to $Q \vee R$ then to $S \vee R$ and then to $S \vee S$. The clausal form of this proposition is S that gives directly a refutation with the clause $\neg S$. Again, each step of the rewriting derivation corresponds to a resolution step in the first presentation, but unlike the search for a refutation derivation, rewriting is deterministic when the rewrite system is confluent.

These examples are straightforward as the propositions involved in the derivations contain no variables. But, if we want to prove the proposition

$$\exists x\ \exists y\ ((a + b) + (x + e) = a + ((b + y) + (d + e)))$$

using the associativity axiom, we may rewrite it into

$$\exists x\ \exists y\ (a + (b + (x + e)) = a + (b + (y + (d + e))))$$

but we cannot build a refutation from the clauses

$$\neg(a + (b + (x + e)) = a + (b + (y + (d + e))))$$

$$z = z$$

since the standard unification algorithm does not "see" that instantiating x by $c + d$ and y by c permits a new reduction of the proposition. This problem has been solved by G. Plotkin [Plotkin, 1972] by the introduction of *equational unification*. This idea led a decade later to the so-called *equational resolution* and more generally to *theory resolution* [Stickel, 1985]. This idea has also been exploited in tableaux methods (see, for instance, [Gallier et al., 1989, Degtyarev and Voronkov, 2001]).

Although they are not unifiable in the ordinary sense, these propositions are equationally unifiable. Informally, for an equational theory $\mathcal{E}$, a $\mathcal{E}$-unification problem consists of equations of the form $t_1 = t_2$ and a solution of such an equation is a substitution σ such that σt_1 and σt_2 are equivalent modulo $\mathcal{E}$.

Equational unification [Jouannaud and Kirchner, 1991] may be undecidable or quite complex. But fortunately deciding unifiability can be postponed and modularized by the use of constraints. Starting from the seminal work of G. Huet on higher-order resolution [Huet, 1972, Huet, 1973], the notion of deduction with constraints spread with constraint logic programming [Jaffar and Lassez, 1987] and then constraint programming. The counterparts in theorem proving are deduction with constraints [Kirchner et al., 1990, Bürckert, 1990, Bürckert, 1991] and complete constraint saturation processes [Nieuwenhuis and Rubio, 1994, Vigneron, 1995, Bachmair et al., 1995]. The integration of rewrite based techniques and orderings in first-order theorem proving surveyed in [Kirchner, 1998] has led to very powerful results and systems. This allows, in combination with equational constraints, to solve problems considered by mathematicians as hard [McCune, 1997] (see also [Colata, 1996]).

Besides this idea of building-in part of the equality in general refutation processes, the same concern of building-in part of the equality in equational reasoning itself [Knuth and Bendix, 1970] has led to the study of equational reasoning modulo, whose main landmarks are the study of associative-commutative completion [Peterson and Stickel, 1981], the general study of coherence of an equational theory with respect to a rewrite system [Jouannaud and Kirchner, 1986], its unified presentation in [Bachmair, 1987] and its extension in [Marché, 1994, Viry, 2002].

Extending narrowing and resolution

Equational resolution permits to integrate smoothly, and in a complete way, term rewriting steps and deduction steps. But it is not enough to handle rules rewriting propositions.

For instance, the theory of integral domains contains the axiom

$$\forall x\ \forall y\ (x \times y = 0 \Leftrightarrow (x = 0 \vee y = 0))$$

This proposition can be turned into a rewrite rule

$$x \times y = 0 \rightarrow x = 0 \vee y = 0$$

This rule rewrites an atomic proposition to a disjunction and again, as far as we know, there is no way to replace it by a rule rewriting terms.

Using this rule, we can prove the proposition

$$\exists z\ (a \times a = z \Rightarrow a = z)$$

but we cannot derive the empty clause from the clausal form of its negation. Indeed with the clauses

$$a \times a = z \qquad \neg(a = z')$$

we cannot apply the resolution rule.

So, when we have such computation rules rewriting propositions, we introduce another proof search rule, called **Extended Narrowing**, that, in this example, compares the proposition $a \times a = z$ with the left-hand side of the rule

$$x \times y = 0 \to x = 0 \vee y = 0$$

and suggests the instantiation $z \mapsto 0$, that permits to reduce the instantiated proposition $a \times a = 0$ to $a = 0 \vee a = 0$ whose clausal form is $a = 0$. Then we can conclude the refutation by a resolution step with the other clause $\neg(a = z')$.

Together with the resolution rule, this extended narrowing rule forms a proof search method, that we call *Extended Narrowing and Resolution* (ENAR). When neither rewrite rules nor equational axioms are considered, the ENAR method is just usual resolution. When we have rewrite rules or equational axioms applying to terms only, the ENAR method is just equational resolution, where the resolution rule generates constraints that are solved with an equational unification algorithm. The new case is when we have also rules rewriting atomic propositions such that the **Extended Narrowing** applies. We prove that, in this case, the ENAR method is complete provided the rewrite system satisfies some properties.

We consider only rules whose left-hand side is an atomic proposition. Hence we do not allow rules of the form $P \wedge Q \to R$. Indeed in the sequent calculus $P \wedge Q$ can be proved using the deduction rules of conjunction. But this possibility would be lost when reducing this proposition to R. Hence the deduction and computation rules would conflict in a severe way as it does in equational deduction modulo, where the problem is solved by coherence restrictions [Jouannaud and Kirchner, 1986]. In the same way, putting the proposition $(P \wedge Q) \vee S$ in clausal form would dispatch the literals P and Q in two different clauses, making it more complicated to retrieve the redex.

This restriction to atomic left-hand sides nevertheless allows for powerful first-order theories like higher-order logic as detailed in section 6.

Sequent calculus modulo

The idea of separating computation steps and deduction steps came to our attention from automated theorem proving. Rewriting techniques have also been implemented in many proof assistants. However, in our opinion, and as also mentioned by [Barendregt and Barendsen, 2002] under the name of *Poincaré principle*, it is of wider interest and deserves to be studied also from a proof theoretical point of view. Indeed, rewriting atomic propositions introducing connectives and quantifiers can change the logical structure of the propositions and hence allows new deduction steps. This leads to a real interaction between computation and deduction and to a non-trivial proof theory. In this paper we define a presentation of *deduction modulo* based on the sequent calculus. Of course, other formalisms such as natural deduction or Hilbert's systems can be used as well. In deduction modulo, the right rule of the conjunction cannot be expressed as usual

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash P \wedge Q, \Delta}$$

because we have to take into account the fact that the conclusion need not be equal to $P \wedge Q$ but may be only congruent to this proposition. Hence this rule is stated

$$\frac{\Gamma \vdash P, \Delta \quad \Gamma \vdash Q, \Delta}{\Gamma \vdash R, \Delta} \text{ if } R \text{ and } P \wedge Q \text{ are congruent}$$

The completeness of the ENAR method

Completeness of *Extended Narrowing And Resolution* is obtained proof-theoretically, relating its derivations to those of the sequent calculus modulo. An alternative semantical method has been recently developed by [Stuber, 2001], under different hypotheses.

As usual in syntactic proofs, completeness is obtained from cut elimination (sometimes under the form of Herbrand's theorem). Here we need cut elimination for the sequent calculus modulo. It depends on the considered congruence, unlike cut elimination for first-order sequent calculus, that is proved once and for all. Cut elimination for various congruences has been proved in [Dowek and Werner, 1999], including all the congruences presented by a confluent and terminating quantifier-free rewrite system, those presented by a confluent and terminating positive rewrite system and that of a first-order presentation of higher-order logic.

It is worth mentioning that there is trivially no ENAR derivation of the contradiction $\bot$ modulo the congruence defining higher-order logic. Hence the completeness of ENAR modulo this congruence implies the consistency of higher-order logic and it cannot be proved in higher-order logic. Thus it is not surprising that powerful proof theoretical results are needed to obtain completeness of this method (for a discussion on this point see [Dowek, 2000]). This completeness theorem suggests new applications of proof theory to automated theorem proving.

Applications

Deduction modulo has many important applications like the integration of decision procedures in theorem provers.

Another important application is the mechanization of higher-order logic. Since we indeed started from this last problem, we exemplify the approach by applying the previous results to a first-order presentation of higher-order logic based on combinators, and we show that the ENAR method subsumes higher-order resolution [Huet, 1972, Huet, 1973]. We show in [Dowek et al., 2001] how this could be apply to a first-order presentation of higher-order logic based on a calculus of explicit substitutions.

Outline of the paper

Section 1 presents the *sequent calculus modulo* and its main properties. Section 2 defines the *Extended Narrowing And Resolution* (ENAR) proof search method and sections 3, 4 and 5 are dedicated to the completeness proof of this method. We conclude in Section 6 by presenting an application to higher-order logic.

History

This paper appeared first as INRIA report 3400 [Dowek et al., 1998] in April 1998 and it is expected to appear by the end of 2003 in the Journal of Automated Reasoning.
This report corresponds to the journal version with more detailled comments and fully expanded proofs where typically only the main proof arguments are presented in the journal version.

1 The sequent calculus modulo

1.1 Definitions

We consider a set of function symbols, usually denoted $f, g, h, \ldots$, a set of predicate symbols, and an infinite set of variables $\mathcal{X}$ which elements are denoted $x, y, z, \ldots$. These sets are assumed to be disjoint. Each function symbol and predicate symbol has a fixed arity. Nullary function symbols are called *constants*. We assume that there is at least one constant. The set of *terms*, *atomic propositions*, *propositions* and *sentences* (i.e. closed propositions) are defined as usual in first-order logic [Gallier, 1986]. We assume that the reader is familiar with the basic notions of rewriting as defined, for instance in [Dershowitz and Jouannaud, 1990]. Since we consider rules rewriting propositions that contain binders (quantifiers), we also use some notions of *combinatory reduction systems* [Klop et al., 1993]. The usual notions such as that of *occurrence* are generalized to propositions, clauses, ... The standard capture avoiding substitution of the term t for the variable x in a proposition P is written $\{t/x\}P$. In contrast, $\{x \mapsto t\}P$ denotes the syntactic replacement of t for x in P. The *domain* of a substitution σ is the set of variables that are not trivially mapped to themselves by σ, we denote it $\mathcal{D}om(\sigma)$. When ω is an occurrence in a proposition P, we write $P_{|\omega}$ for the term or proposition at occurrence ω and $P[t]_\omega$ for the proposition obtained by replacing $P_{|\omega}$ by t in P.

Definition 1.1 *A* term rewrite rule *is a pair of terms $l \to r$. As usual, the variables of r must occur in l. An* equational axiom *is a pair of terms $l = r$. A* proposition rewrite rule *is a pair of propositions $l \to r$, where l is an atomic proposition and r is arbitrary. As usual, the free variables of r must occur in l.*

An example of a term rewrite rule is $x + 0 \to x$ that is part of the theory of groups. An example of an equational axiom is $x + y = y + x$ that is part of the theory of Abelian groups. At last, an example of a proposition rewrite rule is $x \in \wp(y) \to \forall z\ (z \in x \Rightarrow z \in y)$ that is part of set theory.

A *class rewrite system* is a pair, denoted by $\mathcal{RE}$, consisting of:

- $\mathcal{R}$: a set of proposition rewrite rules,
- $\mathcal{E}$: a set of term rewrite rules and equational axioms.

Definition 1.2 *Given a proposition rewrite system $\mathcal{R}$, the proposition P* $\mathcal{R}$-rewrites *to P', denoted $P \rightarrow_{\mathcal{R}} P'$, if $P_{|\omega} = \sigma(l)$ and $P' = P[\sigma(r)]_\omega$, for some rule $l \rightarrow r \in \mathcal{R}$, some occurrence ω in P and some substitution σ. As usual when applying σ, quantified variables of r are renamed to avoid captures.*

For instance, with the rules

$$\begin{array}{rcl} x \times y = 0 & \rightarrow & x = 0 \vee y = 0 \\ a \in \mathcal{P}(b) & \rightarrow & \forall x.(x \in a \Rightarrow x \in b) \end{array}$$

we have the following rewrite steps:

$$\begin{array}{rcl} \forall z\ z \times z = 0 & \rightarrow_{\mathcal{R}} & \forall z\ (z = 0 \vee z = 0) \\ \forall z\ \forall z'\ z \times z' = 0 & \rightarrow_{\mathcal{R}} & \forall z\ \forall z'\ (z = 0 \vee z' = 0) \\ \forall x.(\{y | y = 2 * x\} \in \mathcal{P}(b)) & \rightarrow_{\mathcal{R}} & \forall x(\forall u\ u \in \{y | y = 2 * x\} \Rightarrow u \in B) \end{array}$$

Notice that the proposition $z \times z' = z''$ cannot be rewritten by these rules.

The reflexive-transitive closure of the relation $\rightarrow_{\mathcal{R}}$ is written $\rightarrow^*_{\mathcal{R}}$. The relations $=_{\mathcal{E}}$ and $=_{\mathcal{RE}}$ are the congruences generated respectively by $\mathcal{E}$ and $\mathcal{R} \cup \mathcal{E}$. We then define the notion of $\mathcal{RE}$-rewriting.

Definition 1.3 *Given a class rewrite system $\mathcal{RE}$, the proposition P* $\mathcal{RE}$-rewrites *to P', denoted $P \longrightarrow_{\mathcal{RE}} P'$, if $P =_{\mathcal{E}} Q$, $Q_{|\omega} = \sigma(l)$ and $P' =_{\mathcal{E}} Q[\sigma(r)]_\omega$, for some rule $l \rightarrow r \in \mathcal{R}$, some proposition Q, some occurrence ω in Q and some substitution σ.*

The next two results are direct consequences of the standard rewriting concepts [Jouannaud and Kirchner, 1986, Baader and Nipkow, 1998] and [Terese (M. Bezem, J. W. Klop and R. de Vrijer, eds.), 2002].

Proposition 1.1 *When $\mathcal{E}$ contains only rewrite rules and the rewrite system $\mathcal{R} \cup \mathcal{E}$ is confluent then the relation $\longrightarrow_{\mathcal{RE}}$ is also confluent.*

Proposition 1.2 *If $P =_{\mathcal{RE}} Q$ then $P =_{\mathcal{E}} Q$ or $P \longleftrightarrow^*_{\mathcal{RE}} Q$, i.e. P and Q are related by the reflexive, symmetric and transitive closure of the relation $\longrightarrow_{\mathcal{RE}}$.*

1.2 The sequent calculus modulo

We are now able to give, in Figure 1, the definition of the sequent calculus modulo. This sequent calculus extends the usual one (see, for instance, [Gallier, 1986, Girard et al., 1989]), by allowing working modulo $\mathcal{RE}$. In these rules, Γ and Δ are finite multisets of propositions. Notice that the deduction rule **axiom** requires the left proposition and the right one to be identical modulo the congruence and not just unifiable. *Hence the free variables of a sequent cannot be instantiated and are treated as constants.*

When the congruence $=_{\mathcal{RE}}$ is simply the identity, this sequent calculus collapses to the usual one. In that case sequents are written as usual with the $\vdash$ symbol.

The next two propositions result directly from the definitions:

$$\frac{}{P \vdash_{\mathcal{RE}} Q}\text{axiom}\ \ \text{if } P =_{\mathcal{RE}} Q$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} \Delta}\text{cut}\ \ \text{if } P =_{\mathcal{RE}} Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{contr-l}\ \ \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} Q_1, Q_2, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{contr-r}\ \ \text{if } P =_{\mathcal{RE}} Q_1 =_{\mathcal{RE}} Q_2$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\text{weak-l}$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}\text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\wedge\text{-l}\ \ \text{if } R =_{\mathcal{RE}} (P \wedge Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\wedge\text{-r}\ \ \text{if } R =_{\mathcal{RE}} (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\vee\text{-l}\ \ \text{if } R =_{\mathcal{RE}} (P \vee Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\vee\text{-r}\ \ \text{if } R =_{\mathcal{RE}} (P \vee Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta \quad \Gamma, Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\Rightarrow\text{-l}\ \ \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q)$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\Rightarrow\text{-r}\ \ \text{if } R =_{\mathcal{RE}} (P \Rightarrow Q)$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} P, \Delta}{\Gamma, R \vdash_{\mathcal{RE}} \Delta}\neg\text{-l}\ \ \text{if } R =_{\mathcal{RE}} \neg P$$

$$\frac{\Gamma, P \vdash_{\mathcal{RE}} \Delta}{\Gamma \vdash_{\mathcal{RE}} R, \Delta}\neg\text{-r}\ \ \text{if } R =_{\mathcal{RE}} \neg P$$

$$\frac{}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}\bot\text{-l}\ \ \text{if } P =_{\mathcal{RE}} \bot$$

$$\frac{\Gamma, \{t/x\}Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, x, t)\ \forall\text{-l}\ \ \text{if } P =_{\mathcal{RE}} \forall x\ Q$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \{c/x\}Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, x, c)\ \forall\text{-r}\ \ \text{if } P =_{\mathcal{RE}} \forall x\ Q \text{ and } c \text{ fresh constant}$$

$$\frac{\Gamma, \{c/x\}Q \vdash_{\mathcal{RE}} \Delta}{\Gamma, P \vdash_{\mathcal{RE}} \Delta}(Q, x, c)\ \exists\text{-l}\ \ \text{if } P =_{\mathcal{RE}} \exists x\ Q \text{ and } c \text{ fresh constant}$$

$$\frac{\Gamma \vdash_{\mathcal{RE}} \{t/x\}Q, \Delta}{\Gamma \vdash_{\mathcal{RE}} P, \Delta}(Q, x, t)\ \exists\text{-r}\ \ \text{if } P =_{\mathcal{RE}} \exists x\ Q$$

Figure 1: The *sequent calculus modulo*

Proposition 1.3 *If $=_{\mathcal{RE}}$ is a decidable congruence, then proof checking for the sequent calculus modulo is decidable. This is in particular the case when the rewrite relation $\longrightarrow_{\mathcal{RE}}$ is confluent and (weakly) terminating.*

Proposition 1.4 *If $P =_{\mathcal{RE}} Q$ then $\Gamma \vdash_{\mathcal{RE}} P, \Delta$ if and only if $\Gamma \vdash_{\mathcal{RE}} Q, \Delta$ and $\Gamma, P \vdash_{\mathcal{RE}} \Delta$ if and only if $\Gamma, Q \vdash_{\mathcal{RE}} \Delta$ and the proofs have the same size.*

Notice that in the left rule of the existential quantifier and the right rule of the universal quantifier, we introduce a fresh constant (extending the language) rather than a fresh variable. This does not change the provability relation because free variables are treated as constants in the sequent calculus, but it permits to have the property that if a closed sequent (i.e. a sequent where all the propositions are closed i.e. without free variable) has a proof, then it has a closed proof, (i.e. a proof where all the sequents are closed).

Since all the variables free in a proof can be substituted by a constant, we have:

Proposition 1.5 *If a closed sequent $\Gamma \vdash_{\mathcal{RE}} \Delta$ has a proof, then it also has a proof where all the sequents are closed.*

1.3 The equivalence between $\vdash$ and $\vdash_{\mathcal{RE}}$

The first important result proved about sequent calculus modulo is the equivalence lemma which states the soundness and the completeness of the sequent calculus modulo with respect to first-order logic. A proposition P is provable in the sequent calculus modulo if and only if it is provable in the usual sequent calculus using an appropriate set of axioms. We prove that for each congruence $\mathcal{RE}$, there is a set of axioms $\mathcal{T}$ such that

$$\mathcal{T}, \Gamma \vdash \Delta \text{ if and only if } \Gamma \vdash_{\mathcal{RE}} \Delta.$$

Hence the theorems are the same in the two formalisms. But, of course the proofs are very different: the proofs modulo are shorter because the standard deduction steps performing computations described by $\mathcal{RE}$ are eliminated.

Definition 1.4 *A set of axioms $\mathcal{T}$ and a class rewrite system $\mathcal{RE}$ are said to be* compatible *if:*

- *$P =_{\mathcal{RE}} Q$ implies $\mathcal{T} \vdash P \Leftrightarrow Q$.*
- *for every proposition P in $\mathcal{T}$, we have $\vdash_{\mathcal{RE}} P$.*

Proposition 1.6 *For every class rewrite system $\mathcal{RE}$, there exists a set of axioms $\mathcal{T}$ such that $\mathcal{T}$ and $\mathcal{RE}$ are compatible.*

Proof: For each pair of propositions P and Q such that $P =_{\mathcal{RE}} Q$, we take the proposition

$$\forall x_1 \ \ldots \ \forall x_n \ (P \Leftrightarrow Q)$$

where $x_1, \ldots, x_n$ are the free variables of $P \Leftrightarrow Q$. ◇

Notice that when the language contains an equality predicate and the corresponding axioms, we have a smaller set $\mathcal{T}$ consisting of the axioms

$$\forall x_1 \ \ldots \ \forall x_n \ (l = r)$$

where $x_1, \ldots, x_n$ are the free variables of $l = r$, for each term rewrite rule $l \to r$ and equational axiom $l = r$ and of the proposition

$$\forall x_1 \ \ldots \ \forall x_n \ (l \Leftrightarrow r)$$

where $x_1, \ldots, x_n$ are the free variables of $l \Leftrightarrow r$, for each proposition rewrite rule $l \to r$.

When we do not have an equality predicate, we may add this predicate and the corresponding axioms in a conservative way as detailed in [Dowek, 1999].

Proposition 1.7 *Let $\mathcal{RE}$ be a class rewrite system and $\mathcal{T}$ be a set of axioms such that $\mathcal{T}$ and $\mathcal{RE}$ are compatible. Then we have:*

$$\mathcal{T}, \Gamma \vdash \Delta \quad \textit{if and only if} \quad \mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta.$$

Proof: The "only if" part is obvious since a derivation of $\mathcal{T}, \Gamma \vdash \Delta$ is also a derivation of $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$.

For the "if" part, notice first that, as well known, the **axiom** rule can be replaced by the following alternative rule:

$$\frac{}{\Gamma, P \vdash_{\mathcal{RE}} Q, \Delta}\text{axiom} \quad \text{if } P =_{\mathcal{RE}} Q$$

Then, using the contraction rule, any proof of $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$ can be transformed into a proof where the propositions of $\mathcal{T}$ appear in the left-hand part of every sequent. Then, the proof proceeds by induction on the structure of the derivation of $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$.

As an example, we give the case of the $\wedge$-r rule: if the proof has the form

$$\frac{\dfrac{\pi}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} P, \Delta} \quad \dfrac{\rho}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} Q, \Delta}}{\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} R, \Delta} \wedge\text{-r} \quad \text{where } R =_{\mathcal{RE}} P \wedge Q$$

by induction hypothesis we have proofs π' and ρ' of $\mathcal{T}, \Gamma \vdash P, \Delta$ and $\mathcal{T}, \Gamma \vdash Q, \Delta$. We first build the proof:

$$\frac{\dfrac{\pi'}{\mathcal{T}, \Gamma \vdash P, \Delta} \quad \dfrac{\rho'}{\mathcal{T}, \Gamma \vdash Q, \Delta}}{\mathcal{T}, \Gamma \vdash P \wedge Q, \Delta} \wedge\text{-r}$$

Then, we have $R =_{\mathcal{RE}} (P \wedge Q)$, thus by the first condition of compatibility, we get $\mathcal{T}, \Gamma \vdash (P \wedge Q) \Rightarrow R$. Since the *modus ponens* is a derived rule of the sequent calculus we can build a proof of $\mathcal{T}, \Gamma \vdash R, \Delta$. $\diamond$

Proposition 1.8 (Equivalence) *If the set of axioms $\mathcal{T}$ and the class rewrite system $\mathcal{RE}$ are compatible then we have:*

$$\mathcal{T}, \Gamma \vdash \Delta \quad \textit{if and only if} \quad \Gamma \vdash_{\mathcal{RE}} \Delta.$$

Proof: By proposition 1.7, the sequent $\mathcal{T}, \Gamma \vdash \Delta$ is provable if and only if the sequent $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$ is. As, by the second condition of compatibility each proposition of $\mathcal{T}$ is $\mathcal{RE}$-provable, the sequent $\mathcal{T}, \Gamma \vdash_{\mathcal{RE}} \Delta$ is provable if and only if the sequent $\Gamma \vdash_{\mathcal{RE}} \Delta$ is. ◇

Notice that the proofs of the propositions 1.7 and 1.8 use the cut rule. These propositions may fail when we drop this rule: there are propositions that have a cut free proof in sequent calculus with axioms but not in sequent calculus modulo.

2 Extended narrowing and resolution

We are now ready to extend the usual resolution method to a method where congruences are built-in.
In the rest of the paper, we assume the relation $\longrightarrow_{\mathcal{RE}}$ to be confluent.
Indeed weak termination is not necessary to establish the soundness and correctness of the ENAR proof search method.

2.1 Labels

As in any resolution based proof search method, the first step is the transformation of the propositions to be proved into a set of clauses that involves some skolemization steps. There are several possibilities to skolemize a proposition such that $\forall x\ \exists y\ P(0, y)$ where the quantified variable x does not occur: we can either take the Skolem constant f be nullary $(P(0, f))$ or unary $(P(0, f(x)))$. We have chosen the latter, since in this case, even if we have an equation $x \times 0 = 0$ yielding the $\mathcal{E}$-equivalence of $\forall x\ \exists y\ P(x \times 0, y)$ and $\forall x\ \exists y\ P(0, y)$, the Skolem symbols introduced in both cases have the same arity.

To implement this choice, we have to memorize during the clausal form computation, the universal quantifier scope of each subformula, for instance that the subformula $\exists y\ P(0, y)$ occurs in the scope of $\forall x$. This is done by associating to each subformula a set of variables called a *label*. A pair P^l formed with a proposition P and a label l is called a *labeled proposition*.

Notice that using the same equation, we could replace 0 by $0 \times z$ in the proposition $P(0, f(x))^x$ getting $P(0 \times z, f(x))^x$. In the sequent calculus, the two propositions $\forall x, P(0, f(x))$ and $\forall x, P(0 \times z, f(x))^x$ are equivalent and z is considered as a constant. We could also replace $P(0, f(x)^x$ by $P(0 \times z, f(x)^{x,z}$, asking for the equivalence of the propositions $\forall x, P(0, f(x))$ and $\forall z, \forall x, P(0 \times z, f(x))$. Fortunately, the two equivalences are valid (see the definition of the set of axioms $\mathcal{T}$). But, in our proof search method, there is no need to introduce this variable z. Indeed, the only effect would be to extend the proof

search domain, a point which is not needed as our method is complete without it (as shown further). Thus, we forbid introduction of new variables by equational steps.

Thus, in the definitions below, we constraint all free variables of a labeled proposition to appear in its label, and two $\mathcal{E}$-equivalent labeled propositions to have the same label. This allows, for instance, to have a uniform treatment of labels in the lemma 4.6.

Definition 2.1 *A* labeled proposition *is a pair P^l formed with a proposition P and a finite set l of variables containing all the free variables of P and called its* label.

Definition 2.2 *When we apply a substitution θ to a labeled proposition, we replace each variable x of the label by the free variables of θx. Two labeled propositions P^l and $Q^{l'}$ are $\mathcal{E}$-equivalent if $P =_\mathcal{E} Q$ and $l = l'$. The labeled proposition P^l $\mathcal{R}$-rewrites into $Q^{l'}$ if P $\mathcal{R}$-rewrites to Q (c.f. definition 1.2) and $l = l'$.*

2.2 Constrained Clauses

Clauses and constrained clauses are the main objects processed by resolution proof search algorithms.

Definition 2.3 *A* clause *is a finite set $\{P_1, \ldots, P_n\}$ of labeled propositions such that every P_i is a* literal, *i.e. either an* atomic proposition *or the negation of an atomic proposition. The* empty clause *is denoted* $\Box$.

Let ψ be a set of labeled propositions and P^l a labeled proposition, we write ψ, P^l for the set $\psi \cup \{P^l\}$. Let Φ be a set of sets of labeled propositions and ψ a set of labeled propositions, we write Φ, ψ for the set $\Phi \cup \{\psi\}$.

Definition 2.4 (Clausal form) *We consider the following transformations on sets of sets of labeled propositions.*

- $\Phi, (\psi, (P \wedge Q)^l) \longrightarrow \Phi, (\psi, P^l), (\psi, Q^l)$
- $\Phi, (\psi, (P \vee Q)^l) \longrightarrow \Phi, (\psi, P^l, Q^l)$
- $\Phi, (\psi, (P \Rightarrow Q)^l) \longrightarrow \Phi, (\psi, (\neg P)^l, Q^l)$
- $\Phi, (\psi, \bot^l) \longrightarrow \Phi, \psi$
- $\Phi, (\psi, (\forall x\ P)^l) \longrightarrow \Phi, (\psi, P^{l,x})$ *where x is a fresh variable*
- $\Phi, (\psi, (\exists x\ P)^{y_1,\ldots,y_n}) \longrightarrow \Phi, (\psi, (\{f(y_1, \ldots, y_n)/x\}P)^{y_1,\ldots,y_n})$ *where f is a fresh function symbol*
- $\Phi, (\psi, (\neg(P \wedge Q))^l) \longrightarrow \Phi, (\psi, (\neg P)^l, (\neg Q)^l)$
- $\Phi, (\psi, (\neg(P \vee Q))^l) \longrightarrow \Phi, (\psi, (\neg P)^l), (\psi, (\neg Q)^l)$

- $\Phi, (\psi, (\neg(P \Rightarrow Q))^l) \longrightarrow \Phi, (\psi, P^l), (\psi, (\neg Q)^l)$
- $\Phi, (\psi, (\neg\forall x\ P)^{y_1,\ldots,y_n}) \longrightarrow \Phi, (\psi, (\neg\{f(y_1,\ldots,y_n)/x\}P)^{y_1,\ldots,y_n})$ *where f is a fresh function symbol*
- $\Phi, (\psi, (\neg\exists x\ P)^l) \longrightarrow \Phi, (\psi, (\neg P)^{l,x})$ *where x is a fresh variable*
- $\Phi, (\psi, (\neg\bot)^l) \longrightarrow \Phi$
- $\Phi, (\psi, (\neg\neg\ P)^l) \longrightarrow \Phi, (\psi, P^l)$

To put a set of non labeled sentences in clausal form we first label them with an empty set.

Proposition 2.1 *This transformation terminates and the normal forms of this system are sets of clauses called* clausal forms.

Proof: Each rule decreases the complexity defined on a set of sets of propositions Φ as the multiset of pairs $\langle a, b\rangle$, where a is the number of occurrences of the symbols $\wedge$, $\vee$, $\Rightarrow$, $\bot$, $\forall$, $\exists$ and b the number of occurrences of the symbol $\neg$ for each set in Φ. ◇

Notation As this ordering on sets of sets of propositions is used several times in the following, we name it *ClF-ordering.*

Notation If Φ is a set of sets of labeled propositions, we write $c\ell(\Phi)$ for its clausal form. If $\psi = \{P_1^{l_1}, \ldots, P_n^{l_n}\}$ is a set of labeled propositions then we write also $c\ell(\psi)$ or $c\ell(P_1^{l_1}, \ldots, P_n^{l_n})$ for $c\ell(\{\{P_1^{l_1}\}, \ldots, \{P_n^{l_n}\}\})$, leaving the labels implicit when there is no ambiguity.

Definition 2.5 *For some equational theory $\mathcal{E}$, an* equation *modulo $\mathcal{E}$ (for short an equation) is a pair of terms or of atomic propositions denoted $t =^?_{\mathcal{E}} t'$. A substitution σ is a* $\mathcal{E}$-solution *of $t =^?_{\mathcal{E}} t'$ when $\sigma(t) =_{\mathcal{E}} \sigma(t')$. It is a $\mathcal{E}$-solution of an* equation system *(i.e. of a conjunction of equations) $\mathcal{C}$ when it is a solution of all the equations in $\mathcal{C}$.*

A constrained clause *is a pair $U\,[\mathcal{C}]$ such that U is a clause and $\mathcal{C}$ is a set of equations, called constraints. It schematizes the set of all instances of U by the solutions of $\mathcal{C}$.*

For a given set of constrained clauses $\mathcal{K}$, we call fresh variant *of a clause a renaming of this clause with variables that do not previously occur in the set of clauses. Of course, renaming is performed consistently on the labels.*

This definition can be extended to more general constraints, in particular including ordering constraints on terms, see [Kirchner et al., 1990]. This permits also to improve the efficiency of the ENAR approach, as demonstrated in [Stuber, 2001].

Definition 2.6 (Fresh Variant) *Let ξ be the finite substitution $\{x_1 \mapsto y_1, \ldots, x_n \mapsto y_n\}$ where $y_1, \ldots, y_n$ are distinct variables. Then ξ is a* renaming *away from the set of variables W if the domain and the range of ξ are disjoint, and $W \cap \{y_1, \ldots, y_n\} = \emptyset$.*

For a given set of constrained clauses $\mathcal{K}$, we call fresh variant *of a clause a renaming of this clause with variables that do not previously occur in the set of clauses i.e. a fresh variant of $U\,[\mathcal{C}]$ is $\xi U\,[\xi\mathcal{C}]$ where ξ is a renaming away from the free variables of $\mathcal{K}$. This renaming also affects the variables occurring in the labels.*

Extended Resolution

$$\frac{\{P_1, \ldots, P_n, Q_1, \ldots, Q_m\}\,[\mathcal{C}_1] \quad \{\neg R_1, \ldots, \neg R_p, S_1, \ldots, S_q\}\,[\mathcal{C}_2]}{\{Q_1, \ldots, Q_m, S_1, \ldots, S_q\}\,[\mathcal{C}_1 \cup \mathcal{C}_2 \cup \{P_1 =^?_{\mathcal{E}} \ldots =^?_{\mathcal{E}} P_n =^?_{\mathcal{E}} R_1 \ldots =^?_{\mathcal{E}} R_p\}]}$$

Extended Narrowing

$$\frac{U\,[\mathcal{C}]}{U'\,[\mathcal{C} \cup \{U_{|\omega} =^?_{\mathcal{E}} l\}]} \text{ if } l \to r \in \mathcal{R},\ U_{|\omega} \text{ atom. prop. and } U' \in c\ell(\{U[r]_\omega\})$$

Figure 2: *Extended narrowing and resolution (ENAR)*

2.3 The ENAR method

Definition 2.7 *Let $\mathcal{RE}$ be a class rewrite system and $\mathcal{K}$ be a set of constrained clauses, we write*

$$\mathcal{K} \underset{\mathcal{RE}}{\mapsto\!\!\!\!\!\rightarrow} U\,[\mathcal{C}]$$

if the constrained clause $U\,[\mathcal{C}]$ can be deduced from the constrained clauses of $\mathcal{K}$ using finitely many applications of the **Extended Narrowing** *and* **Extended Resolution** *rules described in Figure 2. This means that there exists a derivation of the clause $U\,[\mathcal{C}]$ under the assumptions $\mathcal{K}$, i.e. a sequence $U_1\,[\mathcal{C}_1], \ldots, U_n\,[\mathcal{C}_n]$ such that either $n = 0$ and $U\,[\mathcal{C}]$ is an element of $\mathcal{K}$ or $n \geq 1$, $U_n\,[\mathcal{C}_n] = U\,[\mathcal{C}]$ and each $U_i\,[\mathcal{C}_i]$ is produced by the application of a rule in ENAR to fresh variants of clauses of the set $\mathcal{K} \cup \{U_1\,[\mathcal{C}_1], \ldots, U_{i-1}\,[\mathcal{C}_{i-1}]\}$.*

The first rule, **Extended Resolution**, is a simple extension of the usual resolution rule where, instead of solving the $\mathcal{E}$-unification constraints, they are stored in the constraint part. A similar rule appeared first in [Huet, 1972, Huet, 1973] (where unification is higher-order unification), then in [Kirchner et al., 1990] and [Bürckert, 1991]. The set of constraints $\{P_1 =^?_{\mathcal{E}} \ldots =^?_{\mathcal{E}} P_n =^?_{\mathcal{E}} R_1 \ldots =^?_{\mathcal{E}} R_p\}$ is an abuse of notation for $\{P_1 =^?_{\mathcal{E}} P_2, \ldots, P_1 =^?_{\mathcal{E}} P_n, P_1 =^?_{\mathcal{E}} R_1, \ldots, P_1 =^?_{\mathcal{E}} R_p\}$. Recall that propositions are labeled with variables, but these labels play no role when applying the **Extended Resolution** rule. In particular, they are removed from the constraints part of all constraint clauses.

The **Extended Narrowing** rule is close to that used in basic or constraint narrowing [Hullot, 1980, Nutt et al., 1989] but the difference here is that narrowing is applied to atomic propositions and not to terms. Because atomic propositions can be rewritten into non-atomic ones, the narrowed clause may not be in clausal form anymore and it must be transformed back into clausal form.

When $\mathcal{R}$ is empty the **Extended Narrowing** rule never applies and we get back equational resolution. When $\mathcal{R}$ and $\mathcal{E}$ are both empty we get back resolution. Notice also that ENAR does *not* subsume paramodulation [Peterson, 1983] as **Extended Narrowing** is always performed with the same built-in congruence.

Remark 2.1 *The rules of the ENAR method may be very prolific since all unification problems are delayed as constraints. In an implementation, it seems mandatory to simplify the*

constraints to detect obviously unsolvable constraints. When we have a solved constraint of the form $x =^?_{\mathcal{E}} t$, we may also want to substitute x by t in the constrained clause. We conjecture that such an optimized method is complete.

Remark 2.2 *If we start with $\mathcal{RE}$-normal propositions, provided the system $\mathcal{R}$ is right-normalized, then the ENAR rules produce only normal clauses. However, we may choose to propagate solutions of the constraints and to normalize the clauses obtained this way. We conjecture this strategy to be complete.*

3 The main theorem

3.1 Soundness and completeness of the ENAR method

The main theorem of this paper is that the ENAR method is sound and complete with respect to the sequent calculus modulo.

Theorem 3.1 (Main Theorem) *Let $\mathcal{RE}$ be a class rewrite system such that $\longrightarrow_{\mathcal{RE}}$ is confluent. For all ϕ and ψ sets of sentences and $\neg\phi = \{\neg P | P \in \phi\}$, if $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints then we have the following implications*

$$c\ell(\phi, \neg\psi)\,[\emptyset] \underset{\mathcal{RE}}{\mapsto\!\!\!\!\twoheadrightarrow} \Box\,[\mathcal{C}] \quad \Rightarrow \quad \phi \vdash_{\mathcal{RE}} \psi$$

If the sequent $\phi \vdash_{\mathcal{RE}} \psi$ has a cut free proof then there exists a derivation

$$c\ell(\phi, \neg\psi)\,[\emptyset] \underset{\mathcal{RE}}{\mapsto\!\!\!\!\twoheadrightarrow} \Box\,[\mathcal{C}]$$

Under the same hypothesis, when the cut rule is eliminable in the sequent calculus modulo $\mathcal{RE}$ we have the following corollary

$$\phi \vdash_{\mathcal{RE}} \psi \quad \Leftrightarrow \quad c\ell(\phi, \neg\psi)\,[\emptyset] \;\mapsto\!\!\!\!\twoheadrightarrow_{\mathcal{RE}}\; \Box\,[\mathcal{C}]$$

If $\mathcal{T}$ is a set of axioms compatible with $\mathcal{RE}$, this can be restated as:

$$\mathcal{T}, \phi \vdash \psi \quad \Leftrightarrow \quad c\ell(\phi, \neg\psi)\,[\emptyset] \;\mapsto\!\!\!\!\twoheadrightarrow_{\mathcal{RE}}\; \Box\,[\mathcal{C}]$$

3.2 The role of cut elimination

For the completeness of ENAR, the cut elimination hypothesis is essential. If the congruence is such that sequent calculus modulo does not have the cut elimination property, then the ENAR method may be incomplete. Consider, for instance the congruence generated by the rule (Crabbé's rule)

$$A \to B \wedge \neg A$$

Modulo this congruence, the proposition $\neg B$ has a proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash_{\mathcal{RE}} A}\text{ axiom}}{A, B \vdash_{\mathcal{RE}} A}\text{ weak-l}}{A, B, \neg A \vdash_{\mathcal{RE}}}\ \neg\text{-l}}{A, A \vdash_{\mathcal{RE}}}\ \wedge\text{-l}}{A \vdash_{\mathcal{RE}}}\ \text{contr-l}}{\vdash_{\mathcal{RE}} \neg A}\ \neg\text{-r} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{A \vdash_{\mathcal{RE}} A}\text{ axiom}}{A, B \vdash_{\mathcal{RE}} A}\text{ weak-l}}{A, B, \neg A \vdash_{\mathcal{RE}}}\ \neg\text{-l}}{A, A \vdash_{\mathcal{RE}}}\ \wedge\text{-l}}{A \vdash_{\mathcal{RE}}}\ \text{contr-l}$$

$$\dfrac{\dfrac{\dfrac{}{B \vdash_{\mathcal{RE}} B}\text{ axiom} \qquad \dfrac{B \vdash_{\mathcal{RE}} \neg A}{B \vdash_{\mathcal{RE}} \neg A}\text{ weak-l}}{B \vdash_{\mathcal{RE}} A}\ \wedge\text{-r} \qquad A \vdash_{\mathcal{RE}}}{B \vdash_{\mathcal{RE}}}\ \text{cut}}{\vdash_{\mathcal{RE}} \neg B}\ \neg\text{-r}$$

But it has no cut free proof.

Although, the proposition $\neg B$ is provable, the empty clause cannot be derived with the ENAR method from $c\ell(\neg\neg B)$, i.e. from the clause B. Indeed, starting from the clause B the extended resolution rule cannot be applied, and the extended narrowing rule generates an unsolvable constraint $A = B$.

3.3 Road map of the proof

According to Herbrand's theorem, a set of clauses Φ is refutable in first-order logic if and only if there is a set of instances of clauses of Φ that is refutable by propositional resolution. This theorem can also be stated as the fact that the system formed with the rules **Instantiation** and **Identical Resolution** given in Figure 3 is a complete refutation system for first-order logic.

This completeness result can be lifted to first-order resolution: a proposition refutable by the previous system is also refutable by first-order resolution. This gives a proof-theoretical way to establish the completeness of first-order resolution (see, for instance, [Gallier, 1986]).

Notice that, in the **Identical Resolution** rule, the eliminated atoms in the two clauses need to be identical. Notice also that in this rule, the clauses need not be ground and that the term t in the **Instantiation** rule may also contain variables. At last, in this rule, we can use the substitution without renaming, because there are no bound variables in clauses.

For higher-order logic, such an intermediate system with an instantiation rule and an identical resolution rule has been introduced by P.B. Andrews [Andrews, 1971]. His system contains also additional rules, for instance a conversion rule.

For deduction modulo, we consider a system containing the rules **Identical Resolution** and **Instantiation** together with two more rules called **Conversion** and **Reduction**. We obtain this way a system called *Extended Identical Resolution* (EIR) described in Figure 3.

Definition 3.1 *Given a class rewrite system $\mathcal{RE}$ and a set of clauses $\mathcal{K}$, we write*

$$\mathcal{K} \hookrightarrow_{\mathcal{RE}} U$$

$$\frac{U}{\{x \mapsto t\}U}\textbf{Instantiation}$$

$$\frac{U}{U'}\textbf{Conversion} \quad \text{if } U =_{\mathcal{E}} U'$$

$$\frac{U}{U'}\textbf{Reduction} \quad \text{if } U \rightarrow_{\mathcal{R}} \psi \text{ and } U' \in c\ell(\{\psi\})$$

$$\frac{U, P^{l_1} \quad U', \neg P^{l_2}}{U \cup U'}\textbf{Identical Resolution}$$

Figure 3: Extended Identical Resolution (EIR)

if the clause U can be deduced from the clauses of $\mathcal{K}$ using finitely many applications of the Extended Identical Resolution *(EIR) rules described in Figure 3. This means that there exists a derivation of the clause U under the assumptions $\mathcal{K}$, i.e. a sequence $U_1, \ldots, U_n$ such that either $n = 0$ and U is an element of $\mathcal{K}$ or $n \geq 1$, $U_n = U$ and each U_i is produced by the application of a rule in EIR to clauses of the set $\mathcal{K} \cup \{U_1, \ldots, U_{i-1}\}$.*

Notice that in the **Instantiation** rule, the instantiated variable is replaced in the label by the free variables of the substituted term. In the **Conversion** rule, because of the definition of $\mathcal{E}$-equivalent labeled propositions (definition 2.2), the labels are kept by the transformed propositions, this forbid in particular to introduce free variables in U' that were not present in the labels of U. In the **Reduction** rule, the labels are extended by the clausal form transformation algorithm. In the **Identical Resolution** rule, the eliminated propositions do not need to have the same label.

Notice that in contrast with what happens for the ENAR system, in the EIR method the clauses are not renamed with fresh variables before the application of the rules, and that renaming, when needed, is performed by the **Instantiation** rule.

Notice also that if $U_1, \ldots, U_n$ is a derivation of $\mathcal{K} \hookrightarrow U_n$, then $U_2, \ldots, U_n$ is a derivation of $\mathcal{K} \cup \{U_1\} \hookrightarrow U_n$.

The proof of theorem 3.1 is detailed in the next sections. The road map of the proof consists first to prove the soundness and completeness of the EIR method, then to prove the soundness and completeness of the ENAR method with respect to the EIR. The precise steps are the following:

$$\mathcal{T}, \phi \vdash \psi \overset{\textit{Prop. 1.8}}{\Leftrightarrow} \phi \vdash_{\mathcal{RE}} \psi \overset{\textit{Prop. 4.2}}{\Leftrightarrow} c\ell(\phi \cup \neg\psi) \hookrightarrow \square \overset{\textit{Prop. 5.2}}{\Leftrightarrow} c\ell(\phi \cup \neg\psi)\,[\emptyset] \twoheadrightarrow \square\,[\mathcal{C}]$$

Each step can be shortly explained as follows:

- The first part of the proof (Prop. 1.8) is the already seen equivalence lemma.

- The second one (propositions 4.1 and 4.2) shows that the EIR method is sound and complete with respect to provability in the sequent calculus modulo. The completeness proof requires in particular that the cut rule is eliminable in the sequent calculus modulo $\mathcal{RE}$.
- The last part of the proof (propositions 5.1 and 5.2) is the lifting of the result from EIR to the ENAR method.

The soundness proofs (proposition 4.1 and 5.1) are quite easy, the completeness proof of ENAR relatively to EIR (proposition 5.2) is an extension of the usual lifting lemma. The completeness of the EIR method is non-trivial. Indeed, because we have rules rewriting atomic propositions to non-atomic ones, we need to put the reduced proposition in clausal form on the fly and in particular to skolemize it on the fly. Hence the completeness proof will involve, as we shall see, properties of skolemization.

From all this we deduce that the set of clauses $c\ell(\Gamma \cup \neg\Delta)$ can be refuted by the ENAR method if and only if the sequent $\mathcal{T}, \Gamma \vdash \Delta$ is provable in sequent calculus.

4 Soundness and Completeness of the EIR method

Before entering in the proof of the main result, let us recall the main notations that will be used, except when explicitly stated:

- Terms are denoted $l, r, s, t, \ldots$,
- Propositions are denoted $L, P, Q, R, \ldots$,
- Set of propositions are denoted by $\phi, \psi, \ldots$, and ψ, P is a notation for the set $\psi \cup \{P\}$,
- Set of sets of propositions are denoted $\Phi, \Psi, \ldots$ and we write Φ, ψ for the set $\Phi \cup \{\psi\}$,
- Clauses are denoted $U, V, W, \ldots$,
- The clausal form of Φ is noted $c\ell(\Phi)$,
- When $\psi = \{P_1, \ldots, P_n\}$ is a set of propositions then $c\ell(\psi)$ or $c\ell(P_1, \ldots, P_n)$ denote $c\ell(\{\{P_1\}, \ldots, \{P_n\}\})$,
- Set of clauses are denoted $\mathcal{K}$,
- When $\psi = \{P_1, \ldots, P_n\}$ is a set of propositions, $\overline{\forall}\ \psi$ denotes the proposition $\forall x_1 \ \ldots \ \forall x_p \ (P_1 \vee \ldots \vee P_n)$ where $x_1, \ldots, x_p$ are the free variables of $P_1, \ldots, P_n$, and the proposition $\bot$ if ψ is empty.

4.1 Soundness

We first start with the soundness of the EIR method. The idea in this proof is to translate a clause as the universal closure of the disjunction of its literals and then to prove that if a set of clauses is refutable by the EIR method then its translation is refutable in sequent calculus. The case of the **Reduction** rule that includes dynamical transformation to clausal form requires the correctness of this transformation.

Definition 4.1 *Let $\psi = \{P_1, \ldots, P_n\}$ be a set of labeled propositions, we consider the universal closure of ψ, defined as*

$$\overline{\forall}\psi = \forall x_1 \ \ldots \ \forall x_p \ (P_1 \vee \ldots \vee P_n)$$

where $x_1, \ldots, x_p$ are the variables in the labels of $P_1, \ldots, P_n$ and the proposition $\bot$ if ψ is empty.

We will use the well-known Skolem theorem:

Theorem 4.1 (Skolem)

$$\Gamma, \forall x_1 \ \ldots \ \forall x_n \ \exists y \ P \vdash \Delta \quad \Leftrightarrow \quad \Gamma, \forall x_1 \ \ldots \ \forall x_n \ \{f(x_1, \ldots, x_n)/y\}P \vdash \Delta$$

where f is a fresh symbol.

Let us first prove the correctness of the clausal form computation given in definition 2.4 and already proven to be terminating:

Lemma 4.1 (Correctness of the clausal form) *Let $\psi_1, \ldots, \psi_p$, $\chi_1, \ldots, \chi_q$ be sets of labeled propositions. If, by the clausal form computation of definition 2.4, we have*

$$\{\psi_1, \ldots, \psi_p\} \longrightarrow \{\chi_1, \ldots, \chi_q\}$$

Then

$$\overline{\forall}\chi_1, \ldots, \overline{\forall}\chi_q \vdash \quad \Leftrightarrow \quad \overline{\forall}\psi_1, \ldots, \overline{\forall}\psi_p \vdash$$

Proof: By induction over the length of the derivation. Let us develop the main case of the existential quantifier. Suppose that we have $\psi_1 = Q_1, \ldots, Q_m, (\exists x \ P)^{y_1, \ldots, y_n}$ and

$$\begin{array}{ll} \chi_1 & = Q_1, \ldots, Q_m, (\{f(y_1, \ldots, y_n)/x\}P)^{y_1, \ldots, y_n} \\ \chi_2 & = \psi_2, \ \ldots, \ \chi_q = \psi_p \quad (p = q \text{ in this case}) \end{array}$$

Let $z_1, \ldots, z_k$ be the variables in the labels of the propositions of ψ_1 but not among $y_1, \ldots, y_n$. Because all the variables free in $\exists x \ P$ are among $y_1, \ldots, y_n$, no variable of $z_1, \ldots, z_k$ is free in P. Hence

$$\overline{\forall}\psi_1 \quad \Leftrightarrow \quad \forall y_1 \ \ldots \ \forall y_n \ (\exists x \ P \vee \forall z_1 \ \ldots \ \forall z_k \ (Q_1 \vee \ldots \vee Q_m))$$

Hence

$$\overline{\forall}\psi_1, \ldots, \overline{\forall}\psi_p \vdash$$

if and only if

$$\forall y_1 \ \ldots \forall y_n \ (\exists x \ P \vee \forall z_1 \ \ldots \ \forall z_k \ (Q_1 \vee \ldots \vee Q_m)), \overline{\forall}\psi_2, \ldots, \overline{\forall}\psi_p \vdash$$

and, by Skolem's theorem, if and only if

$$\forall y_1 \ \ldots \ \forall y_n (P\{f(y_1, \ldots, y_n)/x\} \vee \forall z_1 \ \ldots \ \forall z_k \ (Q_1 \vee \ldots \vee Q_m)),$$
$$\overline{\forall}\psi_2, \ldots, \overline{\forall}\psi_p \vdash$$

if and only if

$$\overline{\forall}\chi_1, \ldots, \overline{\forall}\chi_q \vdash$$

For the case of the universal quantifier, it directly follows from the fact that $\overline{\forall}(\forall x P^{y_1,\ldots,y_n})$ iff $\overline{\forall} P^{y_1,\ldots,y_n,x}$. ◇

Lemma 4.2 *Let* $\{U_1, \ldots, U_n\}$ *be a set of clauses. If* $U_1, \ldots, U_n \hookrightarrow \Box$ *then* $\overline{\forall}U_1, \ldots, \overline{\forall}U_n \vdash_{\mathcal{RE}}$.

Proof: We reason by induction on the structure of the derivation $U_1, \ldots, U_n \hookrightarrow \Box$.

If the derivation is empty, then one of the clauses U_i is $\Box$. Thus, the proposition $\overline{\forall}\ U_i$ is $\bot$ and $\overline{\forall}U_1, \ldots, \overline{\forall}U_n \vdash_{\mathcal{RE}}$.

Otherwise, the derivation of $U_1, \ldots, U_n \hookrightarrow \Box$ starts by producing a clause U' and there is a shorter derivation of $U_1, \ldots, U_n, U' \hookrightarrow \Box$.

By induction hypothesis, we have

$$\overline{\forall}U_1, \ldots, \overline{\forall}U_n, \overline{\forall}U' \vdash_{\mathcal{RE}}$$

We consider the rule used to produce U'.

- If this rule is **Reduction**, then there is a clause, say U_1, that reduces to ψ and $U' \in c\ell(\{\psi\})$. The sequent $\overline{\forall}U_1 \vdash_{\mathcal{RE}} \overline{\forall}\psi$ is provable, since $\overline{\forall}U_1 \rightarrow_{\mathcal{R}} \overline{\forall}\psi$ and by definition of the axiom rule.
  Let $\{U', U'_1, \ldots, U'_q\} = c\ell(\{\psi\})$. We have a proof of the sequent

  $$\overline{\forall}U_1, \ldots, \overline{\forall}U_n, \overline{\forall}U' \vdash_{\mathcal{RE}}$$

  and by weakening, we build one of the sequent

  $$\overline{\forall}U_1, \ldots, \overline{\forall}U_n, \overline{\forall}U', \overline{\forall}U'_1, \ldots, \overline{\forall}U'_q \vdash_{\mathcal{RE}}$$

  Let $\mathcal{T}$ be a set of axioms compatible with $\mathcal{RE}$ in the sense of definition 1.4. We have

  $$\{\mathcal{T}, U_1, \ldots, U_n, \psi\} \longrightarrow \{\mathcal{T}, U_1, \ldots, U_n, U', U'_1, \ldots, U'_q\}$$

Thus, by lemma 4.1, we get a proof of the sequent

$$\mathcal{T}, \overline{\forall} U_1, \ldots, \overline{\forall} U_n, \overline{\forall} \psi \vdash$$

and therefore by proposition 1.8, of

$$\overline{\forall} U_1, \ldots, \overline{\forall} U_n, \overline{\forall} \psi \vdash_{\mathcal{RE}}$$

As the sequent $\overline{\forall} U_1 \vdash_{\mathcal{RE}} \overline{\forall} \psi$ is provable, we get the result using the cut rule and structural rules.

- If this rule is **Identical Resolution** then, there are two clauses, say U_1 and U_2 such that
$$U_1 = U_1', P, \quad U_2 = U_2', \neg P, \quad U' = U_1' \cup U_2'$$
The sequent $\overline{\forall} U_1, \overline{\forall} U_2 \vdash_{\mathcal{RE}} \overline{\forall} U'$ is provable and so is the sequent
$$\overline{\forall} U_1, \ldots, \overline{\forall} U_n, \overline{\forall} U' \vdash_{\mathcal{RE}}$$
And we get the result using the cut rule and structural rules.
- If this rule is **Conversion** then there is a clause, say U_1 that is convertible to U'. The sequent $\overline{\forall} U_1 \vdash_{\mathcal{RE}} \overline{\forall} U'$ is provable and we conclude as in the previous case.
- If this rule is **Instantiation**, then there is a clause, say U_1 such that $U' = \{x \mapsto t\} U_1$. The sequent $\overline{\forall} U_1 \vdash_{\mathcal{RE}} \overline{\forall} U'$ is provable and we conclude as in the previous case.

◇

Proposition 4.1 (EIR Soundness)
Let $P_1, \ldots, P_n, Q_1, \ldots, Q_m$ be sentences. If

$$c\ell(P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \square$$

then

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

Proof: Let $\{U_1, \ldots, U_p\}$ be $c\ell(P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m)$. We have

$$U_1, \ldots, U_p \hookrightarrow \square$$

Hence by lemma 4.2, $\overline{\forall} U_1, \ldots, \overline{\forall} U_p \vdash_{\mathcal{RE}}$ i.e. $\mathcal{T}, \overline{\forall} U_1, \ldots, \overline{\forall} U_p \vdash$, where $\mathcal{T}$ is a set of axioms compatible with $\mathcal{RE}$. As

$$\{\mathcal{T}, \{P_1\}, \ldots, \{P_n\}, \{\neg Q_1\}, \ldots, \{\neg Q_m\}\} \longrightarrow \{\mathcal{T}, U_1, \ldots, U_p\}$$

we get by correctness of clausal form computation (lemma 4.1),

$$\mathcal{T}, P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m \vdash$$

i.e.

$$P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m \vdash_{\mathcal{RE}}$$

and finally

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

◇

We could try to sharpen this soundness result and prove that if a sequent is provable by the EIR method, then it has a cut free proof. With the completeness result below, we would have an exact characterization of the sequents provable by the EIR method as those that have a cut free proof. This is not considered in this paper.

4.2 Completeness

We now consider the completeness proof of EIR (proposition 4.2) and first present the structure and main ideas of the proof.

The goal here is to prove that if a sequent $\Gamma \vdash_{\mathcal{RE}} \Delta$ has a cut free proof in deduction modulo, then $c\ell(\Gamma, \neg\Delta) \hookrightarrow \Box$.

Because we have rewrite rules transforming atomic propositions into non-atomic ones, the transformation to clausal form (including skolemization) cannot be done only as an initial part of the algorithm but also each time the **Reduction** rule is used.

Thus if we try to prove that, when $U_1, \ldots, U_n$ are clauses, $(\overline{\forall} U_1, \ldots, \overline{\forall} U_n \vdash_{\mathcal{RE}})$ implies $(U_1, \ldots, U_n \hookrightarrow \Box)$ the recursion does not go through, because sentences that are not clauses may occur in the proof of $\overline{\forall} U_1, \ldots, \overline{\forall} U_n \vdash_{\mathcal{RE}}$. Thus we have to prove directly that

$$\Gamma \vdash_{\mathcal{RE}} \Delta$$

implies

$$c\ell(\Gamma, \neg\Delta) \hookrightarrow \Box$$

The proof proceeds by induction on the size of the cut free proof of $\Gamma \vdash_{\mathcal{RE}} \Delta$ and by cases on the last rule.

Let assume for instance that the last rule is $\wedge$-r

$$\frac{\Gamma \vdash_{\mathcal{RE}} B, \Delta \quad \Gamma \vdash_{\mathcal{RE}} C, \Delta}{\Gamma \vdash_{\mathcal{RE}} A, \Delta}$$

where $A =_{\mathcal{RE}} B \wedge C$ and $\Gamma \vdash_{\mathcal{RE}} B, \Delta$ and $\Gamma \vdash_{\mathcal{RE}} C, \Delta$ have shorter proofs.

We first use the lemma 4.8 below to prove that A $\mathcal{RE}$-reduces to a conjunction $B' \wedge C'$ and that $B' =_{\mathcal{RE}} B$ and $C' =_{\mathcal{RE}} C$. Then using proposition 1.4 we obtain proofs of the

same size for $\Gamma \vdash_{\mathcal{RE}} B', \Delta$ and $\Gamma \vdash_{\mathcal{RE}} C', \Delta$. Using the induction hypothesis we get that $c\ell(\Gamma, \neg B', \neg\Delta) \hookrightarrow \Box$ and $c\ell(\Gamma, \neg C', \neg\Delta) \hookrightarrow \Box$.

We construct a derivation of $c\ell(\Gamma, \neg A, \neg\Delta) \hookrightarrow \Box$ out of these two derivations in two steps. The first constructs a derivation of $c\ell(\Gamma, \neg(B' \wedge C'), \neg\Delta) \hookrightarrow \Box$ and the second uses the lemma 4.6 below to transform it into a derivation of $c\ell(\Gamma, \neg A, \neg\Delta) \hookrightarrow \Box$.

All the cases of the proof are treated similarly, but, the construction of the derivation depends on the used rule. Four cases require special lemmas. The cases $\vee$-l and $\wedge$-r are handled by the lemma 4.7 that is standard in syntactic completeness proofs of resolution.

The cases $\forall$-l and $\exists$-r are handled by lemma 4.5 and are more involved. For instance, if we have a proof of the form:

$$\frac{\Gamma, \forall y\ \exists z\ P(t,y,z) \vdash_{\mathcal{RE}} \Delta}{\Gamma, \forall x\ \forall y\ \exists z\ P(x,y,z) \vdash_{\mathcal{RE}} \Delta}\ \forall\text{-l}$$

then the clausal form of the proposition $\forall x\ \forall y\ \exists z\ P(x,y,z)$ is $P(x,y,f(x,y))$ with a binary Skolem symbol f, while that of the proposition $\forall y\ \exists z\ P(t,y,z)$ is $P(t,y,g(y))$ with a unary Skolem symbol g. In such a case, we have to build a refutation of $P(x,y,f(x,y))$ from one of $P(t,y,g(y))$, i.e. replace any subterm of the form $g(u)$ by $f(t,u)$.

To achieve this goal we need to introduce a notion of substitution of a function symbol. To distinguish such a substitution from others, we shall call it a *transformation* of a function symbol. Properties of transformations are given by two additional lemmas 4.3 and 4.4 that are related to Skolem's theorem and are proved using techniques also used in syntactical proofs of Skolem's theorem.

We are now ready to enter the completeness proof.

Definition 4.2 (Transformation of a function symbol) *Let t be a term (resp. a proposition), f a function symbol of arity n and u a term whose free variables are among $x_1, \ldots, x_n$. The* individual transformation of symbol f *into u is denoted by $(x_1, \ldots, x_n)u/f$. Its application on a term (resp. proposition) t, denoted by $\{(x_1, \ldots, x_n)u/f\}t$, is obtained by replacing in t any subterm of the form $f(v_1, \ldots, v_n)$, where $v_1, \ldots, v_n$ are arbitrary terms, by the term $\{x_1 \mapsto v_1, \ldots, x_n \mapsto v_n\}u$.*

For a finite set of indexes I, we define the result of the application of a transformation of function symbols $\gamma = \{(x_1^i, \ldots, x_n^i)u^i/f^i\}_{i\in I}$ *to a term (resp. a proposition) t as the simultaneous application of the individual symbol transformations on t.*

The labels are not affected by such transformations.

Lemma 4.3 *Let Φ be a set of sets of labeled propositions and γ a transformation of function symbols. We assume that the Skolem symbols introduced when putting Φ in clausal form are fresh, i.e. not transformed by γ. Then, we have $c\ell(\gamma\Phi) = \gamma c\ell(\Phi)$ up to some renaming.*

Proof: We check that if Φ and Φ' are sets of sets of propositions such that $\Phi \longrightarrow \Phi'$ for the rewrite system of definition 2.4, then $\gamma\Phi \longrightarrow \gamma\Phi'$. For instance if

$$\Phi = \Phi_0, (\psi, (\exists x\ P)^{y_1,\ldots,y_p})$$

and

$$\Phi' = \Phi_0, (\psi, (\{f(y_1, \ldots, y_p)/x\}P)^{y_1,\ldots,y_p})$$

then the set

$$\gamma\Phi = \gamma\Phi_0, (\gamma\psi, (\exists x\ \gamma P)^{y_1,\ldots,y_p})$$

transforms to

$$\gamma\Phi_0, (\gamma\psi, (\{f(y_1, \ldots, y_p)/x\}\gamma P)^{y_1,\ldots,y_p}) = \gamma\Phi'$$

since we assume that f is a fresh function symbol.

The result follows by induction on the length of the transformation of Φ to its clausal form. ◇

Lemma 4.4 (A Skolem Lemma for EIR) *Let $\mathcal{K}$ be a set of sets of clauses and γ a transformation of function symbols not appearing in $\mathcal{RE}$. If $\mathcal{K} \hookrightarrow \Box$ then $\gamma\mathcal{K} \hookrightarrow \Box$ and the derivations have the same length.*

Proof: By induction on the structure of the derivation.
For **Instantiation** we have $\gamma\{x \mapsto v\}U = \{x \mapsto \gamma v\}\gamma U$.
For **Conversion** and **Reduction**, if $U =_{\mathcal{E}} V$ then $\gamma U =_{\mathcal{E}} \gamma V$ and if $U \longrightarrow_{\mathcal{RE}} \psi$ then $\gamma U \longrightarrow_{\mathcal{RE}} \gamma\psi$ as the symbols transformed by γ do not appear in $\mathcal{RE}$.
Finally the case of **Identical Resolution** is clear. ◇

Lemma 4.5 *Let t be a closed term, x a variable, $\mathcal{K}$ a set of clauses, and ψ a set of labeled propositions, then*

$$\mathcal{K} \cup c\ell(\{\{t/x\}\psi\}) \hookrightarrow \Box \quad \Rightarrow \quad \mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$$

Proof: Following the structure of the clause form transformation, we proceed by Noetherian induction on the ClF-ordering defined in notation 2.2.

If all the propositions of ψ are literals, then ψ is a clause, $c\ell(\{\psi\}) = \{\psi\}$ and $c\ell(\{\{t/x\}\psi\}) = \{\{t/x\}\psi\}$. With the **Instantiation** rule, we can derive $\{t/x\}\psi$ from ψ. Hence, if $\mathcal{K} \cup c\ell(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

Otherwise, there is a proposition P in ψ that is not a literal. We write $\psi = \psi' \cup \{P\}$ and we detail the different cases.

- If $P = Q_1 \wedge Q_2$ then $\{t/x\}P = \{t/x\}Q_1 \wedge \{t/x\}Q_2$.

$$c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{Q_1\}\}) \cup c\ell(\{\psi' \cup \{Q_2\}\})$$

and

$$\begin{aligned} c\ell(\{\{t/x\}\psi\}) = \quad & c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q_1\}) \cup \\ & c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q_2\}) \end{aligned}$$

Thus, if $\mathcal{K} \cup cl(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then

$$\mathcal{K} \cup cl(\{\{t/x\}\psi' \cup \{\{t/x\}Q_1\}) \cup cl(\{\{t/x\}\psi' \cup \{\{t/x\}Q_2\}) \hookrightarrow \Box$$

By induction hypothesis, we have

$$\mathcal{K} \cup cl(\{\{t/x\}\psi' \cup \{\{t/x\}Q_1\}) \cup cl(\{\psi' \cup \{Q_2\}) \hookrightarrow \Box$$

and using the induction hypothesis a second time

$$\mathcal{K} \cup cl(\{\psi' \cup \{Q_1\}) \cup cl(\{\psi' \cup \{Q_2\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup cl(\{\psi\}) \hookrightarrow \Box$.

- If $P = Q_1 \vee Q_2$ then $\{t/x\}P = \{t/x\}Q_1 \vee \{t/x\}Q_2$.

  $$cl(\{\psi\}) = cl(\{\psi' \cup \{Q_1, Q_2\}\})$$

  and

  $$cl(\{\{t/x\}\psi\}) = cl(\{\{t/x\}\psi' \cup \{\{t/x\}Q_1, \{t/x\}Q_2\}\})$$

  Thus, if $\mathcal{K} \cup cl(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then

  $$\mathcal{K} \cup cl(\{\{t/x\}\psi' \cup \{\{t/x\}Q_1, \{t/x\}Q_2\}\}) \hookrightarrow \Box$$

  By induction hypothesis, we have

  $$\mathcal{K} \cup cl(\{\psi' \cup \{Q_1, Q_2\}\}) \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup cl(\{\psi\}) \hookrightarrow \Box$.

- If $P = \bot$ then $cl(\{\psi\}) = cl(\{\psi'\})$ and $cl(\{\{t/x\}\psi\}) = cl(\{\{t/x\}\psi'\})$. Thus, if $\mathcal{K} \cup cl(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then

  $$\mathcal{K} \cup cl(\{\{t/x\}\psi'\}) \hookrightarrow \Box$$

  By induction hypothesis, we have

  $$\mathcal{K} \cup cl(\{\psi'\}) \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup cl(\{\psi\}) \hookrightarrow \Box$.

- If $P = \neg\bot$ then $cl(\{\psi\}) = \emptyset$ and $cl(\{\{t/x\}\psi\}) = \emptyset$. Thus, if

  $$\mathcal{K} \cup cl(\{\{t/x\}\psi\}) \hookrightarrow \Box$$

  then

  $$\mathcal{K} \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup cl(\{\psi\}) \hookrightarrow \Box$.

- If $P = \forall z\ Q$ then $\{t/x\}P = \forall z\ (\{t/x\}Q)$.

$$c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{Q\}\})$$

and

$$c\ell(\{\{t/x\}\psi\}) = c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q\}\})$$

Thus, if $\mathcal{K} \cup c\ell(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then

$$\mathcal{K} \cup c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q\}\}) \hookrightarrow \Box$$

By induction hypothesis, we have

$$\mathcal{K} \cup c\ell(\{\psi' \cup \{Q\}\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

- If $P = \exists z\ Q$ then $\{t/x\}P = \exists z\ \{t/x\}Q$.
If x does not appear in the label of P, then it is not free in P and the result is obvious.
Otherwise, let $y_1, \ldots, y_n, x$ be the label of P. We have

$$c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{\{f(y_1, \ldots, y_n, x)/z\}Q\}\})$$

where f is a fresh Skolem symbol. Since t is closed, the label of $\{t/x\}P$ is $y_1, \ldots, y_n$, therefore we have

$$\begin{array}{l} c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}P\}\}) = \\ c\ell(\{\{t/x\}\psi' \cup \{\{g(y_1, \ldots, y_n)/z\}\{t/x\}Q\}\}) \end{array}$$

where g is a fresh Skolem symbol. By hypothesis, we have

$$\mathcal{K} \cup c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}P\}\}) \hookrightarrow \Box$$

which is

$$\mathcal{K} \cup c\ell(\{\{t/x\}\psi' \cup \{\{g(y_1, \ldots, y_n)/z\}\{t/x\}Q\}\}) \hookrightarrow \Box$$

and with $\gamma = \{(y_1, \ldots, y_n)f(y_1, \ldots, y_n, t)/g\}$, by lemma 4.4 and 4.3 we have

$$\begin{array}{ll} & \mathcal{K} \cup c\ell(\gamma(\{\{t/x\}\psi' \cup \{\{g(y_1, \ldots, y_n)/z\}\{t/x\}Q\}\})) \hookrightarrow \Box \\ \Leftrightarrow & \mathcal{K} \cup c\ell(\{\{t/x\}\psi' \cup \{\{f(y_1, \ldots, y_n, t)/z\}\{t/x\}Q\}\}) \hookrightarrow \Box \\ \Leftrightarrow & \mathcal{K} \cup c\ell(\{t/x\}(\{\psi' \cup \{\{f(y_1, \ldots, y_n, x)/z\}Q\}\})) \hookrightarrow \Box \end{array}$$

and by induction hypothesis

$$\mathcal{K} \cup c\ell(\{\psi' \cup \{\{f(y_1, \ldots, y_n, x)/z\}Q\}\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

- If $P = \neg\neg Q$ then $\{t/x\}P = \neg\neg(\{t/x\}Q)$.

  $$c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{Q\}\})$$

  and

  $$c\ell(\{\{t/x\}\psi\}) = c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q\}\})$$

  Thus, if $\mathcal{K} \cup c\ell(\{\{t/x\}\psi\}) \hookrightarrow \Box$ then

  $$c\ell(\{\{t/x\}\psi' \cup \{\{t/x\}Q\}\}) \hookrightarrow \Box$$

  By induction hypothesis, we have

  $$\mathcal{K} \cup c\ell(\{\psi' \cup \{Q\}\}) \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

- The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are similar to the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg\forall z\ Q$ is similar to $\exists z\ Q$. The case $\neg\exists z\ Q$ is similar to $\forall z\ Q$.

$\diamond$

Lemma 4.6 *Let $\psi = \{P_1, \ldots, P_n\}$ and $\chi = \{Q_1, \ldots, Q_n\}$ be two sets of labeled propositions such that for all i, $P_i \longrightarrow^*_{\mathcal{R}\mathcal{E}} Q_i$. Then if $\mathcal{K} \cup c\ell(\chi) \hookrightarrow \Box$ then $\mathcal{K} \cup c\ell(\psi) \hookrightarrow \Box$.*

Proof: We proceed again by induction on ψ with the ClF-ordering.

First, if $P \longrightarrow^*_{\mathcal{R}} Q$ then for each clause V of $c\ell(Q)$ there is a clause U of $c\ell(P)$ such that V can be derived from U by the **Reduction** rule (remember that left-members of R are atomic propositions). Second, if $P =_{\mathcal{E}} Q$ then for each clause V of $c\ell(Q)$ there is a clause U of $c\ell(P)$ such that V can be derived from U by the **Conversion** rule. Thus, if all the propositions of ψ are literals, ψ is a clause, $c\ell(\{\psi\}) = \{\psi\}$ and from $\mathcal{K}, \psi$ we can derive, with the **Reduction** rule and **Conversion**, all the clauses of $c\ell(\{\chi\})$. Hence if $\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$, then $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

Otherwise, recall first that by definition 2.2, $P \longrightarrow^*_{\mathcal{R}\mathcal{E}} Q$ implies that P and Q have the same label. There is a proposition P in ψ that is not a literal and that reduces on a proposition Q. We write $\psi = \psi' \cup \{P\}$, $\chi = \chi' \cup \{Q\}$ and we detail the different cases.

- If $P = R_1 \wedge R_2$ then $Q = R'_1 \wedge R'_2$, $R_1 \longrightarrow^*_{\mathcal{R}\mathcal{E}} R'_1$ and $R_2 \longrightarrow^*_{\mathcal{R}\mathcal{E}} R'_2$.
  We have $c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{R_1\}\}) \cup c\ell(\{\psi' \cup \{R_2\}\})$ and $c\ell(\chi) = c\ell(\{\chi' \cup \{R'_1\}\}) \cup c\ell(\{\chi' \cup \{R'_2\}\})$. Hence, if

  $$\mathcal{K} \cup c\ell(\{\chi' \cup \{R'_1\}\}) \cup c\ell(\{\chi' \cup \{R'_2\}\}) \hookrightarrow \Box$$

then, by induction hypothesis,

$$\mathcal{K} \cup c\ell(\{\chi' \cup \{R_1'\}\}) \cup c\ell(\{\psi' \cup \{R_2\}\}) \hookrightarrow \Box$$

and

$$\mathcal{K} \cup c\ell(\{\psi' \cup \{R_1\}\}) \cup c\ell(\{\psi' \cup \{R_2\}\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup c\ell(\psi) \hookrightarrow \Box$.

- If $P = R_1 \vee R_2$ then $Q = R_1' \vee R_2'$, $R_1 \longrightarrow^*_{\mathcal{RE}} R_1'$ and $R_2 \longrightarrow^*_{\mathcal{RE}} R_2'$.
  We have $c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{R_1, R_2\}\})$ and $c\ell(\chi) = c\ell(\{\chi' \cup \{R_1', R_2'\}\})$. Hence, if

  $$\mathcal{K} \cup c\ell(\{\chi' \cup \{R_1', R_2'\}\}) \hookrightarrow \Box$$

  then, by induction hypothesis,

  $$\mathcal{K} \cup c\ell(\{\psi' \cup \{R_1, R_2\}\}) \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup c\ell(\psi) \hookrightarrow \Box$.
- If $P = \bot$ then $Q = \bot$. We have $c\ell(\{\psi\}) = c\ell(\{\psi'\})$ and $c\ell(\chi) = c\ell(\{\chi'\})$ Hence, if

  $$\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$$

  then

  $$\mathcal{K} \cup c\ell(\{\chi'\}) \hookrightarrow \Box$$

  and, by induction hypothesis,

  $$\mathcal{K} \cup c\ell(\{\psi'\}) \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup c\ell(\psi) \hookrightarrow \Box$.
- If $P = \neg\bot$ then $Q = \neg\bot$. We have $c\ell(\{\psi\}) = \emptyset$ and $c\ell(\chi) = \emptyset$ Hence, if

  $$\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$$

  then

  $$\mathcal{K} \hookrightarrow \Box$$

  i.e. $\mathcal{K} \cup c\ell(\psi) \hookrightarrow \Box$.
- If $P = \forall x\ R$ then $Q = \forall x\ R'$ and $R \longrightarrow^*_{\mathcal{RE}} R'$ and R and R' have the same label.
  We have $c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{R\}\})$ and $c\ell(\{\chi\}) = c\ell(\{\chi' \cup \{R'\}\})$. Hence if

  $$\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$$

  i.e.

  $$\mathcal{K} \cup c\ell(\{\chi' \cup \{R'\}\}) \hookrightarrow \Box$$

  then by induction hypothesis

  $$\mathcal{K} \cup c\ell(\{\psi' \cup \{R\}\}) \hookrightarrow \Box$$

  i.e. $c\ell(\{\psi\}) \hookrightarrow \Box$.

- If $P = \exists x\ R$ then $Q = \exists x\ R'$ and $R \longrightarrow^*_{\mathcal{R}\mathcal{E}} R'$. Let $y_1, \ldots, y_p$ be the common label of P and Q. As we are dealing with two independent refutations, without loss of generality, we can choose the same Skolem symbols in both cases. We have

$$c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{\{f(y_1, \ldots, y_p)/x\}R\}\})$$

and

$$c\ell(\{\chi\}) = c\ell(\{\chi' \cup \{\{f(y_1, \ldots, y_p)/x\}R'\}\})$$

Hence, if

$$\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$$

i.e.

$$\mathcal{K} \cup c\ell(\{\chi' \cup \{\{f(y_1, \ldots, y_p)/x\}R'\}\}) \hookrightarrow \Box$$

then by induction hypothesis

$$\mathcal{K} \cup c\ell(\{\psi' \cup \{\{f(y_1, \ldots, y_p)/x\}R\}\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

- If $P = \neg\neg R$ then: $Q = \neg\neg R'$ and $R \longrightarrow^*_{\mathcal{R}\mathcal{E}} R'$. We have $c\ell(\{\psi\}) = c\ell(\{\psi' \cup \{R\}\})$ and $c\ell(\{\chi\}) = c\ell(\{\chi' \cup \{R'\}\})$ Thus, if

$$\mathcal{K} \cup c\ell(\{\chi\}) \hookrightarrow \Box$$

then

$$c\ell(\{\chi \cup \{R'\}\}) \hookrightarrow \Box$$

By induction hypothesis, we have

$$\mathcal{K} \cup c\ell(\{\psi \cup \{R\}\}) \hookrightarrow \Box$$

i.e. $\mathcal{K} \cup c\ell(\{\psi\}) \hookrightarrow \Box$.

- The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are similar to the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg\forall z\ Q$ is similar to $\exists z\ Q$. The case $\neg\exists z\ Q$ is similar to $\forall z\ Q$.

$\diamond$

Lemma 4.7 *Let R, S be sentences and Γ, Δ set of sentences, if*

$$c\ell(R, \Gamma, \neg\Delta) \hookrightarrow \Box \text{ and } c\ell(S, \Gamma, \neg\Delta) \hookrightarrow \Box$$

then we can build a derivation of:

$$c\ell(R \vee S, \Gamma, \neg\Delta) \hookrightarrow \Box$$

Proof: Let $\Gamma = \{P_2, \ldots, P_n\}$, $\Delta = \{Q_1, \ldots, Q_m\}$, $\mathcal{K}_1 = c\ell(R)$, $\mathcal{K}_2 = c\ell(S)$, $\mathcal{K}_3 = c\ell(R \vee S)$ and $\mathcal{K}_4 = c\ell(P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m)$.

We choose the variables of $\mathcal{K}_1$ and $\mathcal{K}_2$ disjoint but for the variables of $\mathcal{K}_3$ we choose the same as in $\mathcal{K}_1$ and $\mathcal{K}_2$.

We check that:

$$c\ell(R, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) = \mathcal{K}_1 \cup \mathcal{K}_4$$

$$c\ell(S, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) = \mathcal{K}_2 \cup \mathcal{K}_4$$

$$c\ell(R \vee S, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) = \mathcal{K}_3 \cup \mathcal{K}_4$$

Thus $\mathcal{K}_1 \cup \mathcal{K}_4 \hookrightarrow \Box$ and $\mathcal{K}_2 \cup \mathcal{K}_4 \hookrightarrow \Box$.

Now $\mathcal{K}_3 = \{U_1 \cup U_2 \mid U_1 \in \mathcal{K}_1, U_2 \in \mathcal{K}_2\}$. Consider an arbitrary clause U in $\mathcal{K}_2$. We transform the derivation of $\mathcal{K}_1 \cup \mathcal{K}_4 \hookrightarrow \Box$ in such a way that each time we use in $\mathcal{K}_1 \cup \mathcal{K}_4 \hookrightarrow \Box$ a clause U' in $\mathcal{K}_1$ we use the clause $U \cup U'$ instead. We get this way either a derivation of $\mathcal{K}_3 \cup \mathcal{K}_4 \hookrightarrow \Box$ or a derivation of $\mathcal{K}_3 \cup \mathcal{K}_4 \hookrightarrow U$.

If we do not get a derivation of $\mathcal{K}_3 \cup \mathcal{K}_4$ for some clause U of $\mathcal{K}_2$, then for every clause U of $\mathcal{K}_2$ we get a derivation of $\mathcal{K}_3 \cup \mathcal{K}_4 \hookrightarrow U$. We construct a derivation of $\mathcal{K}_3 \cup \mathcal{K}_4 \hookrightarrow \Box$ using these derivations and that of $\mathcal{K}_2 \cup \mathcal{K}_4 \hookrightarrow \Box$. ◇

The next lemma expresses that in the proofs, we can restrict the use of the congruence to reductions.

Lemma 4.8 *If the relation $\longrightarrow_{\mathcal{RE}}$ is confluent we have the following.*

- *If P and $Q \wedge R$ are sentences such that $P =_{\mathcal{RE}} Q \wedge R$, then there exists a sentence $Q' \wedge R'$ such that $P \longrightarrow^*_{\mathcal{RE}} Q' \wedge R'$, $Q =_{\mathcal{RE}} Q'$ and $R =_{\mathcal{RE}} R'$.*
- *If P and $Q \vee R$ are sentences such that $P =_{\mathcal{RE}} Q \vee R$, then there exists a sentence $Q' \vee R'$ such that $P \longrightarrow^*_{\mathcal{RE}} Q' \vee R'$, $Q =_{\mathcal{RE}} Q'$ and $R =_{\mathcal{RE}} R'$.*
- *If P and $Q \Rightarrow R$ are sentences such that $P =_{\mathcal{RE}} Q \Rightarrow R$, then there exists a sentence $Q' \Rightarrow R'$ such that $P \longrightarrow^*_{\mathcal{RE}} Q' \Rightarrow R'$, $Q =_{\mathcal{RE}} Q'$ and $R =_{\mathcal{RE}} R'$.*
- *If P and $\neg Q$ are sentences such that $P =_{\mathcal{RE}} \neg Q$, then there exists a sentence $\neg Q'$ such that $P \longrightarrow^*_{\mathcal{RE}} \neg Q'$ and $Q =_{\mathcal{RE}} Q'$.*
- *If P is a sentence such that $P =_{\mathcal{RE}} \bot$, then $P \longrightarrow^*_{\mathcal{RE}} \bot$.*
- *If P and $\forall x\ Q$ are sentences such that $P =_{\mathcal{RE}} \forall x\ Q$, then there exists a sentence $\forall x\ Q'$ such that $P \longrightarrow^*_{\mathcal{RE}} \forall x\ Q'$ and $Q =_{\mathcal{RE}} Q'$.*
- *If P and $\exists x\ Q$ are sentences such that $P =_{\mathcal{RE}} \exists x\ Q$, then there exists a sentence $\exists x\ Q'$ such that $P \longrightarrow^*_{\mathcal{RE}} \exists x\ Q'$ and $Q =_{\mathcal{RE}} Q'$.*

Proof: Consider, for instance, the case $P =_{\mathcal{RE}} Q \wedge R$. Using proposition 1.2, we either have $P =_{\mathcal{E}} Q \wedge R$ and the result is obvious, or P and Q are related by the reflexive-symmetric-transitive closure of the relation $\longrightarrow_{\mathcal{RE}}$. As this relation is confluent, we have $P \rightarrow^*_{\mathcal{RE}} T$ and $Q \wedge R \rightarrow^*_{\mathcal{RE}} T$. Since $\mathcal{R}$ rewrites only atomic propositions, T has the form $Q' \wedge R'$ with $Q =_{\mathcal{RE}} Q'$ and $R =_{\mathcal{RE}} R'$. $\diamond$

Proposition 4.2 (EIR Completeness)
Assuming the relation $\longrightarrow_{\mathcal{RE}}$ to be confluent, and $P_1, \ldots, P_n, Q_1, \ldots, Q_m$ to be sentences, if the sequent

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

has a cut free proof then:

$$c\ell(P_1, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

Proof: By induction on the size of a closed cut free proof of

$$P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$$

- If the last rule is axiom then $n = m = 1$ and $P_1 =_{\mathcal{RE}} Q_1$. By confluence, there exists sentences R and R' such that $P_1 \longrightarrow^*_{\mathcal{RE}} R$, $Q_1 \longrightarrow^*_{\mathcal{RE}} R'$ and $R =_{\mathcal{E}} R'$. By induction on the structure of R and using the rules **Conversion**, **Instantiation** and **Identical Resolution** we prove that $c\ell(R, \neg R') \hookrightarrow \Box$ and by lemma 4.6 we get
$$c\ell(P_1, \neg Q_1) \hookrightarrow \Box$$
- If the last rule is contr-l or contr-r then the clausal form of the antecedent and the succedent of this rule are the same, thus we simply apply the induction hypothesis. If the last rule is weak-l or weak-r then the clausal form of the antecedent is a subset of the clausal form of the succedent, thus we simply apply the induction hypothesis.
- If the last rule is $\wedge$-l then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to a conjunction $R \wedge S$. By lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} R' \wedge S'$ with $R' =_{\mathcal{RE}} R$, $S' =_{\mathcal{RE}} S$. By induction hypothesis and proposition 1.4:
$$c\ell(R', S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$
and
$$c\ell(R' \wedge S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$
and by lemma 4.6
$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\vee$-l then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to a disjunction $R \vee S$. By lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} R' \vee S'$ and $R' =_{\mathcal{RE}} R$, $S' =_{\mathcal{RE}} S$. By induction hypothesis and proposition 1.4:

$$c\ell(R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

and

$$c\ell(S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

By lemma 4.7 we get:

$$c\ell(R' \vee S', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

and by lemma 4.6

$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\wedge$-r or $\Rightarrow$-l, we proceed as for $\vee$-l. If it is $\vee$-r or $\Rightarrow$-r, we proceed as for $\wedge$-l.
- If the last rule is $\neg$-l then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to a negation $\neg R$. By the lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} \neg R'$ and $R' =_{\mathcal{RE}} R$. By induction hypothesis and proposition 1.4:

$$c\ell(P_2, \ldots, P_n, \neg R', \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

and by lemma 4.6

$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\neg$-r then we proceed as for $\neg$-l.
- if the last rule is $\bot$-l, then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to $\bot$. By lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} \bot$.
  The set $c\ell(\bot, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m)$ contains the clause $\Box$, hence:

$$c\ell(\bot, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

and by lemma 4.6

$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\forall$-l then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to a universal proposition $\forall x\ R$. By lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} \forall x\ R'$ and $R =_{\mathcal{RE}} R'$. By induction hypothesis and proposition 1.4

$$c\ell(\{t/x\}R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

for some closed term t. The labels of P_1, $\forall x\ R'$ and $\{t/x\}R'$ are empty, while that of R' is x. Let us call:

$$\mathcal{K} = c\ell(P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m)$$

By lemma 4.5 $\mathcal{K} \cup c\ell(R') \hookrightarrow \Box$, i.e. $\mathcal{K} \cup c\ell(\forall x\ R') \hookrightarrow \Box$ and by lemma 4.6

$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\exists$-l then one of the P_i's (say P_1) is $\mathcal{RE}$-equivalent to an existential proposition $\exists x\ R$. By lemma 4.8, $P_1 \longrightarrow^*_{\mathcal{RE}} \exists x\ R'$ and $R =_{\mathcal{RE}} R'$. By induction hypothesis and proposition 1.4

$$c\ell(\{c/x\}R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

where c is a fresh constant, i.e.

$$c\ell(\exists x\ R', P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

and by lemma 4.6

$$c\ell(P_1, P_2, \ldots, P_n, \neg Q_1, \ldots, \neg Q_m) \hookrightarrow \Box$$

- If the last rule is $\forall$-r then we proceed as for $\exists$-l and if it is $\exists$-r then we proceed as for $\forall$-l.

$\diamond$

5 Soundness and Completeness of the ENAR method

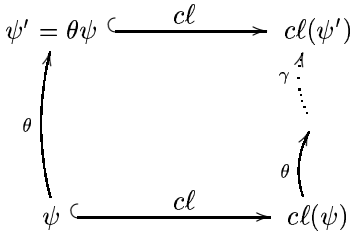
We now lift the soundness and completeness of EIR to ENAR. The two main steps are propositions 5.1 (together with lemma 5.2) where we prove that ENAR derivations can be translated to EIR and proposition 5.2 where we prove conversely that EIR derivations can be translated to ENAR. Again the proofs proceed by induction on the length of derivations.

The main difficulty is to handle the interaction between substitution and clausal form transformation. Lemmas 5.1 and 5.3 state some commutation properties. These lemmas are reminiscent of lemma 4.5 but with different technicalities.

5.1 Soundness

Lemma 5.1 *Let ψ be a set of labeled propositions and θ a closed substitution (i.e. a substitution that maps every variable of its domain to a closed term) such that the variables bound in ψ are not in the domain of θ. Then, there exists a transformation γ of the function*

symbols introduced by putting ψ in clausal form such that $cl(\{\theta\psi\}) = \gamma\theta cl(\{\psi\})$. This can be pictured as follows:



Proof: It is done by induction on ψ with the ClF-ordering. If all the propositions of ψ are literals then

$$cl(\{\theta\psi\}) = \{\theta\psi\} = \theta cl(\{\psi\})$$

we take the identity for γ.

Otherwise there is a proposition P in ψ that is not a literal. We let $\psi = \psi_0 \cup \{P\}$.

- If $P = Q_1 \wedge Q_2$ then by induction hypothesis

$$cl(\{\theta\psi_0 \cup \{\theta Q_1\}\}) = \gamma\theta cl(\{\psi_0 \cup \{Q_1\}\})$$

and

$$cl(\{\theta\psi_0 \cup \{\theta Q_2\}\}) = \gamma'\theta cl(\{\psi_0 \cup \{Q_2\}\})$$

Since the domains of γ and γ' are disjoint (we always assume the Skolem symbols to be fresh),

$$\begin{aligned} cl\{\theta\psi\} &= cl(\{\theta\psi_0 \cup \{\theta Q_1\}\}) \cup cl(\{\theta\psi_0 \cup \{\theta Q_2\}\}) \\ &= (\gamma \cup \gamma')\theta(cl(\{\psi_0 \cup \{Q_1\}\}) \cup cl(\{\psi_0 \cup \{Q_2\}\})) \\ &= (\gamma \cup \gamma')\theta cl(\{\psi\}) \end{aligned}$$

- If $P = Q_1 \vee Q_2$ then by induction hypothesis

$$cl(\{\theta\psi_0 \cup \{\theta Q_1, \theta Q_2\}\}) = \gamma\theta cl(\{\psi_0 \cup \{Q_1, Q_2\}\})$$

i.e.

$$cl(\{\theta\psi\}) = \gamma\theta cl(\{\psi\})$$

- If $P = \bot$ then by induction hypothesis

$$cl(\{\theta\psi_0\}) = \gamma\theta cl(\{\psi_0\})$$

i.e.

$$cl(\{\theta\psi\}) = \gamma\theta cl(\{\psi\})$$

- If $P = \neg\bot$ then we have $cl(\{\theta\psi\}) = \emptyset = \theta cl(\{\psi\})$. We take the identity for γ.

- If $P = \forall x\ Q$ then x is not in the domain of θ and we have $c\ell(\{\theta\psi\}) = c\ell(\{\theta\psi_0 \cup \{\theta Q\}\}) = \gamma\theta c\ell(\{\psi_0 \cup \{Q\}\}) = \gamma\theta c\ell(\{\psi\})$.
- If $P = \exists x\ Q$ then let $y_1, \ldots, y_p$ be the variables of the label of P that are in the domain of θ and $z_1, \ldots, z_q$ the others. The label of P is $y_1, \ldots, y_p, z_1, \ldots, z_q$ and that of θP is $z_1, \ldots, z_q$ because θ is closed.

  We have

  $$c\ell(\{\psi\}) = c\ell(\{\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q\}\})$$
  $$c\ell(\theta\psi) = c\ell(\{\theta\psi_0 \cup \{\{f(z_1, \ldots, z_q)/x\}\theta Q\}\})$$

  Furthermore

  $$\theta\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q = \{g(\theta y_1, \ldots, \theta y_p, z_1, \ldots, z_q)/x\}\theta Q$$

  Taking $\gamma = \{(y_1, \ldots, y_p, z_1, \ldots, z_q) f(z_1, \ldots, z_q)/g\}$ (notice that all the variables of $g(\theta y_1, \ldots, \theta y_p, z_1, \ldots, z_q)\}$ are in $\{z_1, \ldots, z_q\}$) we get

  $$\begin{aligned}\gamma\theta\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q &= \{f(z_1, \ldots, z_q)/x\}\theta Q \\ &= \theta\{f(z_1, \ldots, z_q)/x\}Q\end{aligned}$$

  Thus

  $$\gamma\theta(\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q\}) = \theta(\psi_0 \cup \{\{f(z_1, \ldots, z_q)/x\}Q\})$$

  Hence

  $$\begin{aligned}&c\ell(\{\theta\psi\}) = \\ &c\ell(\{\theta(\psi_0 \cup \{\{f(z_1, \ldots, z_q)/x\}Q\})\}) = \\ &c\ell(\{\gamma\theta(\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q)\})\end{aligned}$$

  By lemma 4.3

  $$c\ell(\{\theta\psi\}) = \gamma c\ell(\theta(\{\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q\}))$$

  and by induction hypothesis

  $$\begin{aligned}&c\ell(\theta(\{\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q\})) = \\ &\gamma'\theta c\ell(\{\psi_0 \cup \{\{g(y_1, \ldots, y_p, z_1, \ldots, z_q)/x\}Q\}) = \\ &\gamma'\theta c\ell(\{\psi\})\end{aligned}$$

  Hence $c\ell(\{\theta\psi\}) = \gamma\gamma'\theta c\ell(\{\psi\})$.
- If $P = \neg\neg Q$ then by induction hypothesis, we have

  $$c\ell(\theta\{\psi_0 \cup \{Q\}\}) = \gamma\theta c\ell(\{\psi_0 \cup \{Q\}\})$$

  i.e. $c\ell(\{\theta\psi\}) = \gamma\theta c\ell(\{\psi\})$.

- The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are similar to the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg\forall z\ Q$ is similar to $\exists z\ Q$. The case $\neg\exists z\ Q$ is similar to $\forall z\ Q$.

$\diamond$

Lemma 5.2 *Let $\mathcal{K}$ be a set of constrained clauses such that*

$$\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\,[\mathcal{C}]$$

and θ be a closed substitution, unifier of $\mathcal{C}$, mapping all the variables of $\mathcal{C}$ to a closed term.

Then, there exists a set $\hat{\mathcal{K}}$ of constrained clauses that are renamings of clauses of $\mathcal{K}$ (each constrained clause of $\mathcal{K}$ can have zero, one or more copies in $\hat{\mathcal{K}}$) such that θ unifies all the constraints of all the clauses of $\hat{\mathcal{K}}$ and $\theta\hat{\mathcal{K}} \hookrightarrow \Box$.

Proof: By induction on the structure of the derivation of

$$\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\,[\mathcal{C}]$$

If this derivation is empty then the constrained clause $\Box\,[\mathcal{C}]$ is an element of $\mathcal{K}$, we take $\hat{\mathcal{K}} = \{\Box\,[\mathcal{C}]\}$.

If the derivation starts by generating a constrained clause U, then the set $\mathcal{K}, U$ has a smaller derivation. By induction hypothesis, there exists a set $\hat{\mathcal{K}}$ of renamings of constrained clauses of $\mathcal{K}$, constrained clauses $U_1, \ldots, U_n$ that are renamings of U such that the substitution θ unifies all the constraints of all the clauses of $\hat{\mathcal{K}}, U_1, \ldots, U_n$ and $\theta\hat{\mathcal{K}}, \theta U_1, \ldots, \theta U_n \hookrightarrow \Box$.We discuss by cases on the rule producing U.

- U is produced by **Extended Resolution** from the renamings of two constrained clauses V and W of $\mathcal{K}$. Then U_i is produced by this same rule from renamings V_i and W_i of V and W. Since all the constraints of V_i and W_i are constraints of U_i they are unified by θ. The clause θU_i is produced by the rules **Conversion** and **Identical Resolution** from the clauses θV_i and θW_i.

  We let $\hat{\mathcal{K}}' = \hat{\mathcal{K}}, V_1, \ldots, V_n, W_1, \ldots, W_n$. The substitution θ is a unifier of all the constraints of all the clauses of $\hat{\mathcal{K}}'$ and $\theta\hat{\mathcal{K}}' \hookrightarrow \Box$.

- U is produced by **Extended Narrowing** from the renaming of a constrained clause V of $\mathcal{K}$. Then U_i is produced by this same rule from a renaming V_i of V. Let $\hat{\mathcal{K}}' = \hat{\mathcal{K}}, V_1, \ldots, V_n$. Since all the constraints of V_i are constraints of U_i they are unified by θ, hence θ is a unifier of all the constraints of all the clauses of $\hat{\mathcal{K}}'$.

  Call ω_i the occurrence of V_i where the **Extended Narrowing** rule is applied and $V_i' = V_i[r]_{\omega_i}$ and $U_i \in c\ell(\{V_i'\})$. We have $\theta(V_i[l]_{\omega_i}) \rightarrow_{\mathcal{R}} \theta V_i'$.

  The constrained clause U_i contains the constraint $V_{i\ |\omega_i} =^?_{\mathcal{E}} l$, hence $\theta V_{i\ |\omega_i} =_{\mathcal{E}} \theta l$, $\theta V_i =_{\mathcal{E}} \theta(V_i[l]_{\omega_i})$. As θl is closed, we can put the same label on these propositions,

thus the clause $\theta V_i[\theta l]_{\omega_i}$ can be derived from the clause θV_i with the **Conversion** rule.

Let θ_1 be the restriction of θ to the variables bound in the propositions V_i' and θ_2 its restriction to the other variables.

By lemma 5.1, there exists a transformation γ_i of the function symbols introduced when putting V_i' in clausal form such that $c\ell(\{\theta_2 V_i'\}) = \gamma_i\theta_2 c\ell(\{V_i'\})$. Hence $c\ell(\{\theta V_i'\}) = \gamma_i\theta_2 c\ell(\{V_i'\})$ and $\theta_1 c\ell(\{\theta V_i'\}) = \gamma_i\theta c\ell(\{V_i'\})$. Thus $\gamma_i\theta U_i \in \theta_1 c\ell(\{\theta V_i'\})$. Hence the clause $\gamma_i\theta U_i$ can be derived from $\theta V_i[\theta l]_{\omega_i}$ with **Reduction** rule and the **Instantiation** rule.

We have $\theta\hat{\mathcal{K}}, \theta U_1, \ldots, \theta U_n \hookrightarrow \Box$, hence by lemma 4.4

$$\gamma_1 \ldots \gamma_n(\theta\hat{\mathcal{K}}, \theta U_1, \ldots, \theta U_n) \hookrightarrow \Box$$

i.e.

$$(\theta\hat{\mathcal{K}}, \gamma_1\theta U_1, \ldots, \gamma_n\theta U_n) \hookrightarrow \Box$$

and thus

$$\theta\hat{\mathcal{K}}, \theta V_1, \ldots, \theta V_n \hookrightarrow \Box$$

i.e. $\theta\hat{\mathcal{K}}' \hookrightarrow \Box$.

◇

Proposition 5.1 (ENAR Soundness)
Let $\mathcal{K}$ be a set of (non-constrained) clauses such that

$$\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\ [\mathcal{C}]$$

where $\mathcal{C}$ is a unifiable set of constraints. Then $\mathcal{K} \hookrightarrow \Box$.

Proof: As the set of constraints $\mathcal{C}$ is unifiable, it has a unifier θ mapping all the variables of $\mathcal{C}$ to a closed term. By lemma 5.2 there is a set $\hat{\mathcal{K}}$ of constrained clauses that are renamings of clauses of $\mathcal{K}$ such that $\theta\hat{\mathcal{K}} \hookrightarrow \Box$. All the clauses of $\theta\hat{\mathcal{K}}$ can be derived from those of $\mathcal{K}$ with the **Instantiation** rule. Hence $\mathcal{K} \hookrightarrow \Box$. ◇

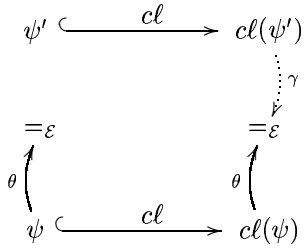
Because of the EIR soundness (proposition 4.1) the previous result entails immediately ENAR soundness.

5.2 Completeness

We now lift the completeness result from EIR to ENAR. To that end we first need a commutation result similar to lemma 5.1.

Lemma 5.3 *Let ψ and ψ' be two sets of labeled propositions. Let θ be a substitution such that no variable bound in ψ is in the domain of θ. Suppose that $\theta\psi =_\mathcal{E} \psi'$. Then there exists a transformation γ of the function symbols introduced by putting ψ' in clausal form such that $\theta c\ell(\{\psi\}) =_\mathcal{E} \gamma c\ell(\{\psi'\})$.*

This can be pictured as follows:



Proof: Notice that ψ' and $\theta\psi$ have the same labels. The proof proceed by induction on ψ using the ClF-ordering.

If all the propositions of ψ are literals then

$$\theta c\ell(\{\psi\}) = \{\theta\psi\} =_\mathcal{E} \{\psi'\} = c\ell(\{\psi'\})$$

we take the identity for γ.

Otherwise there is a proposition P in ψ that is not a literal. We let $\psi = \psi_0 \cup \{P\}$ and $\psi' = \psi'_0 \cup \{P'\}$ with $\theta\psi_0 =_\mathcal{E} \psi'_0$ and $\theta P =_\mathcal{E} P'$.

- If $P = Q_1 \wedge Q_2$ then $P' = Q'_1 \wedge Q'_2$ and $\theta Q_1 =_\mathcal{E} Q'_1$, $\theta Q_2 =_\mathcal{E} Q'_2$. By induction hypothesis
$$\theta c\ell(\{\psi_0 \cup \{Q_1\}\}) =_\mathcal{E} \gamma c\ell(\{\psi'_0 \cup \{Q'_1\}\})$$
and
$$\theta c\ell(\{\psi_0 \cup \{Q_2\}\}) =_\mathcal{E} \gamma' c\ell(\{\psi'_0 \cup \{Q'_2\}\})$$
Since the domains of γ and γ' are disjoint,
$$\begin{array}{ll} \theta c\ell(\{\psi\}) & = \theta c\ell(\{\psi_0 \cup \{Q_1\}\}) \cup \theta c\ell(\{\psi_0 \cup \{Q_2\}\}) \\ & =_\mathcal{E} (\gamma \cup \gamma')(c\ell(\{\psi'_0 \cup \{Q'_1\}\}) \cup c\ell(\{\psi'_0 \cup \{Q'_2\}\})) \\ & = (\gamma \cup \gamma')c\ell(\{\psi'\}) \end{array}$$
- If $P = Q_1 \vee Q_2$ then $P' = Q'_1 \vee Q'_2$ and $\theta Q_1 =_\mathcal{E} Q'_1$, $\theta Q_2 =_\mathcal{E} Q'_2$.
By induction hypothesis
$$\theta c\ell(\{\psi_0 \cup \{Q_1, Q_2\}\}) =_\mathcal{E} \gamma c\ell(\{\psi'_0 \cup \{Q'_1, Q'_2\}\})$$
i.e. $\theta c\ell(\{\psi\}) =_\mathcal{E} \gamma c\ell(\{\psi'\})$.
- If $P = \bot$ then $P' = \bot$. By induction hypothesis
$$\theta c\ell(\{\psi_0\}) =_\mathcal{E} \gamma c\ell(\{\psi'_0\})$$
i.e. $\theta c\ell(\{\psi\}) =_\mathcal{E} \gamma c\ell(\{\psi'\})$.

- If $P = \neg\bot$ then $P' = \neg\bot$. We have $\theta c\ell(\{\psi\}) = \emptyset = c\ell(\{\psi'\})$. We take the identity for γ.
- If $P = \forall x\ Q$ then $P' = \forall x\ Q'$ and $\theta Q =_{\mathcal{E}} Q'$. By induction hypothesis

$$\theta c\ell(\{\psi_0 \cup \{Q\}\}) =_{\mathcal{E}} \gamma c\ell(\{\psi'_0 \cup \{Q'\}\})$$

  i.e. $\theta c\ell(\{\psi\}) =_{\mathcal{E}} \gamma c\ell(\{\psi'\})$.
- If $P = \exists x\ Q$ then $P' = \exists x\ Q'$ and $\theta Q =_{\mathcal{E}} Q'$. Let $y_1, \ldots, y_p$ be the label of P and $z_1, \ldots, z_q$ be the variables free in $\theta y_1, \ldots, \theta y_p$. The label of θP and of P' is $z_1, \ldots, z_q$.
  We have $c\ell(\{\psi\}) = c\ell(\{\psi_0 \cup \{\{f(y_1, \ldots, y_p)/x\}Q\}\})$ and $c\ell(\psi') = c\ell(\{\psi'_0 \cup \{\{g(z_1, \ldots, z_q)/x\}Q'\}\})$.
  As $\theta Q =_{\mathcal{E}} Q'$,

$$\begin{aligned}\theta\{f(y_1, \ldots, y_p)/x\}Q &= \{f(\theta y_1, \ldots, \theta y_p)/x\}\theta Q \\ &=_{\mathcal{E}} \{f(\theta y_1, \ldots, \theta y_p)/x\}Q' \\ &= \gamma\{g(z_1, \ldots, z_q)/x\}Q'\end{aligned}$$

  where $\gamma = \{(z_1, \ldots, z_q)f(\theta y_1, \ldots, \theta y_p)/g\}$. Thus

$$\theta(\psi_0 \cup \{\{f(y_1, \ldots, y_p)/x\}Q\}) =_{\mathcal{E}} \gamma(\psi'_0 \cup \{\{g(z_1, \ldots, z_q)/x\}Q'\})$$

  and by induction hypothesis

$$\begin{aligned}&\theta c\ell(\{\psi_0 \cup \{(\{f(y_1, \ldots, y_p)/x\}Q)\}\}) =_{\mathcal{E}} \\ &\gamma' c\ell(\gamma(\{\psi'_0 \cup \{\{g(z_1, \ldots, z_q)/x\}Q'\}\}))\end{aligned}$$

  and by lemma 4.3

$$\begin{aligned}&\theta c\ell(\{\psi_0 \cup \{(\{f(y_1, \ldots, y_p)/x\}Q)\}\}) =_{\mathcal{E}} \\ &\gamma'\gamma c\ell(\{\psi'_0 \cup \{\{g(z_1, \ldots, z_q)/x\}Q'\}\})\end{aligned}$$

  i.e.

$$\theta c\ell(\{\psi\}) =_{\mathcal{E}} \gamma'\gamma c\ell(\{\psi'\})$$

- If $P = \neg\neg Q$ then $P' = \neg\neg Q'$ and $\theta Q =_{\mathcal{E}} Q'$. By induction hypothesis, we have

$$\theta c\ell(\{\psi_0 \cup \{Q\}\}) =_{\mathcal{E}} \gamma c\ell(\{\psi'_0 \cup \{Q'\}\})$$

  i.e. $c\ell(\{\psi\}) =_{\mathcal{E}} \gamma c\ell(\{\psi'\})$.
- The cases $\neg(Q_1 \vee Q_2)$ and $\neg(Q_1 \Rightarrow Q_2)$ are similar to the case $Q_1 \wedge Q_2$. The cases $\neg(Q_1 \wedge Q_2)$ and $Q_1 \Rightarrow Q_2$ are similar to the case $Q_1 \vee Q_2$. The case $\neg\forall z\ Q$ is similar to $\exists z\ Q$. The case $\neg\exists z\ Q$ is similar to $\forall z\ Q$.

$\diamond$

In the next lemma, we shall use the following local rewriting relation, due to [Peterson and Stickel, 1981]:

Definition 5.1 *The proposition P $\mathcal{R},\mathcal{E}$-*rewrites *to P', denoted $P \longrightarrow_{\mathcal{R},\mathcal{E}} P'$, if $P' = P[\sigma(r)]_\omega$, for some rule $l \to r \in \mathcal{R}$, some occurrence ω in P and some substitution σ such that $\sigma(l) =_\mathcal{E} P_{|\omega}$.*

In the following proposition, to ease reading, a constrained clause $U\,[\mathcal{C}]$ that contains only trivial constraints will be identified with the clause U itself.

Proposition 5.2 (ENAR Completeness) *Let $\mathcal{K}$ be a of constrained clauses, $\hat{\mathcal{K}}$ be a set of constrained clauses that contains renamings of constrained clauses of $\mathcal{K}$, θ be a $\mathcal{E}$-unifier of the constraints of all the clauses of $\hat{\mathcal{K}}$, $\mathcal{J}$ be a set of (non-constrained) clauses such that $\theta\hat{\mathcal{K}} =_\mathcal{E} \mathcal{J}$ and $\mathcal{J} \hookrightarrow \Box$, then $\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\,[\mathcal{C}]$ where $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints.*

Proof: By induction on the length of the derivation of $\mathcal{J} \hookrightarrow \Box$. If the derivation is empty, the set $\mathcal{J}$ contains the empty clause and hence $\mathcal{K}$ contains a clause $\Box\,[\mathcal{C}]$ where $\mathcal{C}$ is a set of constraints and θ (composed with some renaming) is a $\mathcal{E}$-unifier of $\mathcal{C}$.

Otherwise, let $U_1, \ldots, U_n$ be the derivation of $\mathcal{J} \hookrightarrow \Box$. The sequence $U_2, \ldots, U_n$ is a derivation of $\Box$ under the set $\mathcal{J}, U_1$. The clause U_1 has been produced by some EIR rule applied to elements of $\mathcal{J}$. We detail the four cases.

- If the used rule is **Instantiation**, using $x \mapsto t$, then there exists a clause U in $\mathcal{J}$ such that $U_1 = \{x \mapsto t\}U$. There exists a constrained clause $U'\,[\mathcal{C}']$ in $\hat{\mathcal{K}}$ such that $\theta U' =_\mathcal{E} U$. Let $U''\,[\mathcal{C}'']$ be a renaming of $U'\,[\mathcal{C}']$ with fresh variables. Let η be a substitution such that $U'\,[\mathcal{C}'] = \eta(U''\,[\mathcal{C}''])$.

  We have $\{x \mapsto t\}\theta\eta U'' = \{x \mapsto t\}\theta U' =_\mathcal{E} \{x \mapsto t\}U = U_1$. Let $\theta' = (\{x \mapsto t\} \circ \theta \circ \eta)_{|FV(U''[\mathcal{C}''])}$. We have $\theta' U'' =_\mathcal{E} U_1$ and θ' is a unifier of $\mathcal{C}''$.

  The substitutions θ and θ' have disjoint domains. Let $\theta'' = \theta \cup \theta'$. The substitution θ'' is a $\mathcal{E}$-unifier of all the constraints of $\hat{\mathcal{K}}, U''\,[\mathcal{C}'']$, by definition $\theta''(\hat{\mathcal{K}}, U''\,[\mathcal{C}'']) =_\mathcal{E} \mathcal{J}, U_1$ and $\hat{\mathcal{K}}, U''\,[\mathcal{C}'']$ is a set formed with renamings of clauses of $\mathcal{K}$.

  The sequence $U_2, \ldots, U_n$ is a derivation of $\Box$ under the set $\mathcal{J}, U_1$ hence, by induction hypothesis $\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\,[\mathcal{C}]$ where $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints.

- If the used rule is **Conversion**, we have simply

$$\theta\hat{\mathcal{K}} =_\mathcal{E} \mathcal{J} =_\mathcal{E} \mathcal{J}, U_1.$$

  The sequence $U_2, \ldots, U_n$ is a derivation of $\Box$ under the set $\mathcal{J}, U_1$ hence, by induction hypothesis $\mathcal{K} \mapsto\!\!\!\!\!\rightarrow \Box\,[\mathcal{C}]$ where $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints.

- If the used rule is **Identical Resolution**, the set $\mathcal{J}$ contains two clauses that contain opposite literals P and $\neg P$. Thus, in $\hat{\mathcal{K}}$ there are two constrained clauses containing respectively literals $P_1, \ldots, P_n$ and $\neg Q_1, \ldots, \neg Q_p$ such that $\theta P_i =_\mathcal{E}$

$P =_{\mathcal{E}} \theta Q_j$. The **Extended Resolution** rule applied to these clauses leads to a constrained clause $U'[\mathcal{C}']$. The substitution θ is a $\mathcal{E}$-unifier of $\mathcal{C}'$ and $\theta U' =_{\mathcal{E}} U_1$.
Therefore, in $\mathcal{K}$ there are two constrained clauses such that the **Extended Resolution** rule applied to fresh variants of these clauses leads to the constrained clause $U'[\mathcal{C}']$.
The sequence $U_2, \ldots, U_n$ is a derivation of $\Box$ under the set $\mathcal{J}, U_1$ and by induction hypothesis $\mathcal{K}, U'[\mathcal{C}'] \mapsto\!\!\!\!\twoheadrightarrow \Box[\mathcal{C}]$ where $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints. Hence, $\mathcal{K} \mapsto\!\!\!\!\twoheadrightarrow \Box[\mathcal{C}]$.

- If the used rule is **Reduction**, then there is in the set $\mathcal{J}$ a clause V, P, such that $P \rightarrow_{\mathcal{R}} Q$ and $U_1 \in c\ell(\{V \cup \{Q\}\})$. Thus, there exists a constrained clause $(W, P_1, \ldots, P_m)[\mathcal{C}_1]$ in $\hat{\mathcal{K}}$ such that $\theta W =_{\mathcal{E}} V$ and $\theta P_i =_{\mathcal{E}} P$.
We have $\theta P_i =_{\mathcal{E}} P$ and $P \rightarrow_{\mathcal{R}} Q$ thus, as $\mathcal{R}$ applies only to atomic propositions and P_i is a literal, we have $\theta P_i \longrightarrow_{\mathcal{R},\mathcal{E}} Q$. Hence the proposition θP_i contains an occurrence ω_i (that is either the empty occurrence if P is an atomic proposition or 1 if it is the negation of an atomic proposition) such that $(\theta P_i)_{|\omega_i} =_{\mathcal{E}} \sigma_i l$ and $(\theta P_i)[\sigma r]_{\omega_i} = Q$ for some substitution σ_i and rule $l \rightarrow r$ (renaming the variables in the rules, we can choose the domains of the σ_i disjoint and disjoint from all the other variables). Since $\mathcal{R}$ reduces only propositions and θ substitutes variables by terms, the substitution θ is trivially $\mathcal{R},\mathcal{E}$-normal (i.e. the image of all the variables in its domain are in $\mathcal{R},\mathcal{E}$-normal form). Hence the occurrence ω_i is an occurrence of P_i and therefore $(\theta P_i)_{|\omega_i} = \theta(P_{i\ |\omega_i})$. Let $Q_i = (P_i[r]_{\omega_i})$ (i.e. $Q_i = r$ or $Q_i = \neg r$ according to the sign of P_i) and $\mathcal{C}' = \mathcal{C}_1 \cup \bigcup_{i=1}^{n} \{P_{i\ |\omega_i} =^{?}_{\mathcal{E}} l\}$. Let $\theta' = \theta \cup \bigcup_{i=1}^{n} \sigma_i$. Because the domain of the substitutions σ_i contains only fresh variables, the substitution θ' is a $\mathcal{E}$-unifier of $\mathcal{C}'$, $\theta' W = \theta W =_{\mathcal{E}} V$ and

$$\theta' Q_i = \theta'(P_i[r]_{\omega_i}) = \theta'(P_i)[\theta'(r)]_{\omega_i}) = \theta(P_i)[\sigma_i(r)]_{\omega_i} = Q$$

(i.e. $\theta' Q_i = \sigma_i r = Q$ or $\theta' Q_i = \neg\sigma_i r = Q$ according to the sign of P_i). Thus $\theta'(W, Q_1, \ldots, Q_m) =_{\mathcal{E}} V, Q$.
We have $\theta'(W, Q_1, \ldots, Q_m) =_{\mathcal{E}} V, Q$, therefore by lemma 5.3 there is a transformation of function symbols such that $\theta' c\ell(\{W \cup \{Q_1, \ldots, Q_m\}\}) =_{\mathcal{E}} \gamma c\ell(\{V \cup \{Q\}\})$. Hence there is a clause U_1' in $c\ell(\{W \cup \{Q_1, \ldots, Q_m\}\})$ such that $\theta' U_1' =_{\mathcal{E}} \gamma U_1$.
We have $\mathcal{J}, U_1 \hookrightarrow \Box$ hence, by lemma 4.4 $\mathcal{J}, \gamma U_1 \hookrightarrow \Box$. Thus, by induction hypothesis $\hat{\mathcal{K}}, U_1'[\mathcal{C}'] \mapsto\!\!\!\!\twoheadrightarrow \Box[\mathcal{C}]$ for a unifiable set of constraints $\mathcal{C}$.
The **Extended Narrowing** rule applies to $(W, P_1, \ldots, P_m)[\mathcal{C}_1]$ leading, in several steps, to $U_1'[\mathcal{C}']$.
Therefore, $\mathcal{K}$ contains a constrained clause such that the **Extended Narrowing** rule applied to fresh variants of these clauses leads to the constrained clause $U_1'[\mathcal{C}']$ and $\mathcal{K}, U_1'[\mathcal{C}'] \mapsto\!\!\!\!\twoheadrightarrow \Box[\mathcal{C}]$ where $\mathcal{C}$ is a $\mathcal{E}$-unifiable set of constraints. Hence, $\mathcal{K} \mapsto\!\!\!\!\twoheadrightarrow \Box[\mathcal{C}]$.

◇

In case the relation $\longrightarrow_{\mathcal{RE}}$ is confluent, and when $P_1, \ldots, P_n$, $Q_1, \ldots, Q_m$ are sentences such that the sequent $P_1, \ldots, P_n \vdash_{\mathcal{RE}} Q_1, \ldots, Q_m$ has a cut free proof then, completeness of EIR (proposition 4.2) together with the previous result immediately entail the main theorem which proof is thus terminated.

6 A typical example: Higher-order logic

To design a proof search method for higher-order logic, P.B. Andrews [Andrews, 1971] proposes to build-in conversion axioms. Unification is then replaced by unification modulo $\beta\eta$-conversion, usually called higher-order unification [Huet, 1975, Huet, 1976], which is the kernel of higher-order resolution [Huet, 1972, Huet, 1973].

The ENAR method extends trivially to many-sorted first-order logic. This extension can be applied to a presentation of higher-order logic in many-sorted first-order logic. This way, it subsumes higher-order resolution. This result is explained in [Dowek et al., 2001] where we develop a first-order presentation of higher-order logic based on the calculus of explicit substitutions[Abadi et al., 1991, Curien et al., 1996] that is intentionally equivalent to the usual presentation based on λ-calculus.

In this section, we have a more modest goal: to apply ENAR to a simpler first-order formulation of higher-order logic based on combinators. This formulation is not fully equivalent to higher-order logic, because combinators equivalence is weaker than $\beta\eta$-equivalence. However we retrieve equivalence if we add the extensionality axiom to both theories (see, for instance, [Dowek, 1995]).

Expressing higher-order logic as a first-order theory and using a first-order proof search method can be an efficient way to implement higher-order proof search, provided we use the right automated deduction tools for first-order logic.

The sorts of this theory are inductively defined by:

- ι and o are sorts,
- if T and U are sorts then $T \to U$ is a sort.

The language contains the constant symbols:

- $S_{T,U,V}$ of sort $(T \to U \to V) \to (T \to U) \to T \to V$,
- $K_{T,U}$ of sort $T \to U \to T$,
- $\dot{\neg}$ of sort $o \to o$,
- $\dot{\vee}$ of sort $o \to o \to o$,
- $\dot{\forall}_T$ of sort $(T \to o) \to o$,

the binary function symbol:

- $\alpha_{T,U}$ of rank $(T \to U, T)\ U$

and the unary predicate symbol:

- ε of rank (o).

Notice that in this presentation there is a distinction between propositions and objects of type o. Objects of type o are built using the combinators $\dot{\neg}$, $\dot{\vee}$ and $\dot{\forall}_T$ while real propositions are built using the real connectives and quantifiers. If t is a term of type o, the corresponding proposition is $\varepsilon(t)$.

The system $\mathcal{RE}$ consists in:

$$\mathcal{R} = \left\{ \begin{array}{l} \varepsilon(\alpha(\dot{\neg}, x)) \to \neg\varepsilon(x) \\ \varepsilon(\alpha(\alpha(\dot{\vee}, x), y)) \to \varepsilon(x) \vee \varepsilon(y) \\ \varepsilon(\alpha(\dot{\forall}_T, x)) \to \forall y\ \varepsilon(\alpha(x, y)) \end{array} \right.$$

$$\mathcal{E} = \left\{ \begin{array}{l} \alpha(\alpha(\alpha(S, x), y), z) \to \alpha(\alpha(x, z), (y, z)) \\ \alpha(\alpha(K, x), y) \to x \end{array} \right.$$

This system is confluent and strongly terminating [Dowek, 1997] and it enjoys cut elimination [Dowek and Werner, 1999].

Let us see, on an example, the correspondence between a proof in the sequent calculus modulo this theory and ENAR derivations. In higher-order logic the proposition $\exists x\ \varepsilon(x)$ is provable using the reduction $\varepsilon(\alpha(\dot{\neg}, c)) \to \neg\varepsilon(c)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varepsilon(c) \vdash_{\mathcal{RE}} \varepsilon(c)}\ \text{axiom}}{\vdash_{\mathcal{RE}} \varepsilon(c), \neg\ \varepsilon(c)}\ \neg\text{-r}}{\vdash_{\mathcal{RE}} \varepsilon(c), \exists x\ \varepsilon(x)}\ (\varepsilon(x), x, \alpha(\dot{\neg}, c))\ \exists\text{-r}}{\vdash_{\mathcal{RE}} \exists x\ \varepsilon(x), \exists x\ \varepsilon(x)}\ (\varepsilon(x), x, c)\ \exists\text{-r}}{\vdash_{\mathcal{RE}} \exists x\ \varepsilon(x)}\ \text{contr-r}$$

But, the clausal form of the negation of the proposition $\exists x\ \varepsilon(x)$ has only the single clause $\neg\varepsilon(x)$ and thus the resolution rule cannot be applied.

This problem was already met in higher-order resolution [Huet, 1972, Huet, 1973], although in a different form because higher-order logic was not presented as a first-order theory there. It has led to the development of a new rule called **Splitting** which, together with the **Resolution** rule, form a complete system for higher-order logic. This **Splitting** rule generates the different cases of proposition construction allowing to continue the search in a complete way. For example $\neg\varepsilon(x)$ will be split in $\neg\varepsilon(p_1), \ldots, \neg\varepsilon(p_n)$ according to the various possibilities to instantiate x in a minimal way.

An interesting consequence of our approach is that it makes clear that splitting is nothing else than narrowing, allowing to understand higher-order resolution from a new point of view.

Let us see, for example, how the ENAR system finds a proof of the proposition $\exists x\ \varepsilon(x)$, whose negated clause form is $\neg\varepsilon(x)$.

$$\begin{array}{ll} \neg\varepsilon(x)\,[\emptyset] & \rightarrow \textbf{Extended Narrowing} \\ & c\ell(\neg\neg\varepsilon(y))\,[\varepsilon(x) =^? \varepsilon(\alpha(\dot{\neg}, y))] \\ & = \ \varepsilon(y)\,[\varepsilon(x) =^? \varepsilon(\alpha(\dot{\neg}, y))] \\ & \rightarrow \textbf{Extended Resolution} \\ & \Box\,[\varepsilon(x) =^? \varepsilon(\alpha(\dot{\neg}, y)), \varepsilon(x') =^? \varepsilon(y)] \end{array}$$

and since the constraints are trivially satisfiable, this terminates the proof.

7 A Generalization

Because the **Extended Narrowing** rule is a generalization of the **Narrowing** rule used for equational unification, it is tempting to use the same ENAR method with another class rewrite system $\mathcal{R}'\mathcal{E}'$ to solve the constraints themselves. In this case, we consider the constraints as propositions and we take the set $\mathcal{E}$ for the set $\mathcal{R}'$ and the empty set for $\mathcal{E}'$. Indeed, we can do better and get the *Narrowing modulo* method by taking a non-empty set for $\mathcal{E}'$.

This requires to allow rules rewriting terms in $\mathcal{R}'$, while so far, we have considered a rewrite system $\mathcal{R}$ containing only rules rewriting atomic propositions to arbitrary propositions. We can extend the method above to a rewrite system containing also rules rewriting terms to terms provided the system verifies the three assumptions below.

The only difference is in the proof of proposition 5.2 where we have used twice the fact that the system $\mathcal{R}$ reduces only propositions.

First, we have used the fact that as $\mathcal{R}$ reduces only propositions and P is a literal, if $P =_\mathcal{E} P'$ and $P' \rightarrow_\mathcal{R} Q$ then $P \longrightarrow\!\!\!\!\!\rightarrow_{\mathcal{R},\mathcal{E}} Q$. This could be replaced by the fact that if $P =_\mathcal{E} P'$ and $P' \rightarrow_\mathcal{R} Q$ then there is a proposition Q' convertible to Q such that $P \longrightarrow\!\!\!\!\!\rightarrow_{\mathcal{R},\mathcal{E}} Q'$. Now to enforce the same property, we need now a strong coherence property between the relation $\longrightarrow\!\!\!\!\!\rightarrow_{\mathcal{R},\mathcal{E}}$ and $=_\mathcal{E}$:

Assumption 7.1 *If a, a', b and b' are either terms or propositions such that $a =_\mathcal{E} a'$ and $a' \longrightarrow\!\!\!\!\!\rightarrow_{\mathcal{R},\mathcal{E}} b$ then there exists b' such that $a \longrightarrow\!\!\!\!\!\rightarrow_{\mathcal{R},\mathcal{E}}{}^* b'$ and $b =_\mathcal{E} b'$.*

Then, we have used the following fact: in a proposition of the form θP, if there is a redex at the occurrence ω, then it is a proposition occurrence and thus an occurrence of P.

To have the same property, we now need the substitution θ to be $\mathcal{R},\mathcal{E}$-normal. To maintain this normality hypothesis, we need to use the **Instantiation** rule with $\mathcal{R}\mathcal{E}$-normal, and hence $\mathcal{R},\mathcal{E}$-normal terms and to apply the **Reduction** rule on innermost $\mathcal{R}$-redexes only. Then to achieve completeness we must prove that any $\hookrightarrow$-deduction can be transformed into another one where we apply the **Reduction** to innermost $\mathcal{R}$-redexes only. This is possible, for instance, when the innermost $\mathcal{R},\mathcal{E}$-reduction terminates.

Assumption 7.2 *The innermost $\mathcal{R},\mathcal{E}$-reduction terminates.*

At last we need to show that innermostness is preserved by the relation $=_\mathcal{E}$. We denote $\longrightarrow\!\!\!\!\!+_{\mathcal{R},\mathcal{E}}{}^i$ the reduction of an innermost redex.

Assumption 7.3 *If $a =_\mathcal{E} a'$ and $a \longrightarrow\!\!\!\!\!+_{\mathcal{R},\mathcal{E}}{}^i b$ then there exists b' such that $a' \longrightarrow\!\!\!\!\!+_{\mathcal{R},\mathcal{E}}{}^i b'$ and $b =_\mathcal{E} b'$.*

Under these assumptions and the confluence of $\rightarrow_{\mathcal{R}\mathcal{E}}$, we strongly believe that the method above is complete. The assumptions above are similar to those necessary to establish the completeness of narrowing modulo [Kirchner, 1985], which is indeed an instance of the method developed here.

Conclusion

We have presented a sequent calculus that operates in the quotient of the set of propositions modulo a congruence which can equate atomic propositions with non-atomic ones.

We have given a proof search method based on extended narrowing and resolution and proved that it is sound and complete with respect to the sequent calculus modulo, for a large class of congruences. This method extends resolution to cope with deduction modulo. As no specific property of resolution is used in this paper, we believe that other proof search methods, such as the tableaux method, could be extended in a similar way.

When we apply this method to the first-order presentation of higher-order logic above, the **Extended Narrowing** rule specializes exactly to the **Splitting** rule of higher-order resolution [Huet, 1972, Huet, 1973]. The only difference with higher-order resolution is that we are using the combinators S and K and not λ-calculus. Using combinators let unification be only equational unification modulo the axioms S and K and not modulo the axioms β and η as in higher-order resolution. Indeed a first-order presentation of higher-order logic based, not on combinators, but on explicit substitutions simulates exactly higher-order resolution. A first step towards such a result, the expression of higher-order unification as equational unification in the calculus of explicit substitutions, has been achieved in [Dowek et al., 2000]. Giving such a full first-order presentation of higher-order logic using explicit substitutions is done in [Dowek et al., 2001].

We hope that this method will also be useful for theories stronger than higher-order logic in particular for higher-order logic extended with equalities (e.g. associativity and commutativity of some operations) simulating this way equational higher-order resolution. A first step towards such a result, the expression of higher-order equational unification as first-order equational unification in the calculus of explicit substitutions, has been achieved in [Kirchner and Ringeissen, 1997].

We also believe that theorem proving modulo is a general framework allowing to backup the integration of decision procedures (as computational part) and theorem provers (the reasoning part).

For implementations, normalizing clauses seems to be mandatory. We have conjectured that this optimized method is complete.

At last, because the **Extended Narrowing** rule is a generalization of the **Narrowing** rule used for equational unification, it is tempting to use the same ENAR method with another class rewrite system $\mathcal{R}'\mathcal{E}'$ to solve the constraints themselves. In this case, we consider the constraints as propositions and we take the set $\mathcal{E}$ for the set $\mathcal{R}'$ and the empty set for $\mathcal{E}'$. Indeed, we can do better and get *Narrowing modulo* [Kirchner, 1985] by taking a non-empty set for $\mathcal{E}'$.

This requires to allow rules rewriting terms in $\mathcal{R}'$, while so far, we have considered a rewrite system $\mathcal{R}$ containing only rules rewriting atomic propositions to arbitrary propositions. We leave for future work the extension of this method to rewrite systems $\mathcal{R}$ containing also rules rewriting terms to terms.

Acknowledgments: The authors want to thank the anonymous referees of the journal version. Their careful reading and comments helped a lot for the clarification of several points in the paper, Eric Deplagne and Jürgen Stuber for their detailed comments.

References

[Abadi et al., 1991] Abadi, M., Cardelli, L., Curien, P.-L., and Lévy, J.-J. (1991). Explicit substitutions. *Journal of Functional Programming*, 1(4):375–416.

[Andrews, 1971] Andrews, P. B. (1971). Resolution in type theory. *Journal of Symbolic Logic*, 36:414–432.

[Baader and Nipkow, 1998] Baader, F. and Nipkow, T. (1998). *Term Rewriting and all That*. Cambridge University Press.

[Bachmair, 1987] Bachmair, L. (1987). *Proof methods for equational theories*. PhD thesis, University of Illinois, Urbana-Champaign, (Ill., USA). Revised version, August 1988.

[Bachmair et al., 1995] Bachmair, L., Ganzinger, H., Lynch, C., and Snyder, W. (1995). Basic paramodulation. *Information and Computation*, 121(2):172–192.

[Barendregt and Barendsen, 2002] Barendregt, H. and Barendsen, E. (2002). Autartik computations in formal proofs. *Journal of Automated Reasoning*, 28(3):321–336.

[Bürckert, 1990] Bürckert, H.-J. (1990). A resolution principle for clauses with constraints. In Stickel, M. E., editor, *Proceedings 10th International Conference on Automated Deduction, Kaiserslautern (Germany)*, volume 449 of *Lecture Notes in Computer Science*, pages 178–192. Springer-Verlag.

[Bürckert, 1991] Bürckert, H.-J. (1991). *A Resolution Principle for a Logic with Restricted Quantifiers*, volume 568 of *Lecture Notes in Artificial Intelligence*. Springer-Verlag.

[Colata, 1996] Colata, G. (1996). With major math proof, brute computers show flash of reasoning power. *The New York Times*. Tuesday December 10.

[Curien et al., 1996] Curien, P.-L., Hardin, T., and Lévy, J.-J. (1996). Confluence properties of weak and strong calculi of explicit substitutions. *Journal of the ACM*, 43(2):362–397.

[Degtyarev and Voronkov, 2001] Degtyarev, A. and Voronkov, A. (2001). Equality reasoning in sequent-based calculi. In Robinson, A. and Voronkov, A., editors, *Handbook of Automated Reasoning*, volume I, chapter 10, pages 611–706. Elsevier Science.

[Dershowitz and Jouannaud, 1990] Dershowitz, N. and Jouannaud, J.-P. (1990). Rewrite Systems. In van Leeuwen, J., editor, *Handbook of Theoretical Computer Science*, chapter 6, pages 244–320. Elsevier Science Publishers B. V. (North-Holland).

[Dowek, 1995] Dowek, G. (1995). Lambda-calculus, combinators and the comprehension scheme. In Dezani-Ciancaglini, M. and Plotkin, G., editors, *Typed Lambda Calculi and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 154–170. Springer-Verlag.

[Dowek, 1997] Dowek, G. (1997). Proof normalization for a first-order formulation of higher-order logic. In Gunter, E. and Felty, A., editors, *Theorem Proving in Higher Order Logics*, volume 1275 of *Lecture Notes in Computer Science*, pages 105–119. Springer-Verlag.

[Dowek, 1999] Dowek, G. (1999). *La Part du Calcul*. Université de Paris 7. Mémoire d'habilitation.

[Dowek, 2000] Dowek, G. (2000). Axioms vs. Rewrite Rules: From Completeness to Cut Elimination. In Kirchner, H. and Ringeissen, C., editors, *Frontiers of Combining Systems*, volume 1794 of *Lecture Notes in Artificial Intelligence*, pages 62–72. Springer-Verlag.

[Dowek et al., 1998] Dowek, G., Hardin, T., and Kirchner, C. (1998). Theorem proving modulo. Rapport de Recherche 3400, Institut National de Recherche en Informatique et en Automatique. `ftp://ftp.inria.fr/INRIA/publication/RR/RR-3400.ps.gz`.

[Dowek et al., 2000] Dowek, G., Hardin, T., and Kirchner, C. (2000). Higher-order unification via explicit substitutions. *Information and Computation*, 157:183–235.

[Dowek et al., 2001] Dowek, G., Hardin, T., and Kirchner, C. (2001). HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic. *Mathematical Structures in Computer Science*, 11(1):21–45.

[Dowek and Werner, 1999] Dowek, G. and Werner, B. (1999). Proof normalization modulo. In *Types for Proofs and Programs*, volume 1657 of *Lecture Notes in Computer Science*, pages 62–77. Springer-Verlag.

[Gallier et al., 1989] Gallier, J., Raatz, S., and Snyder, W. (1989). Rigid E-unification and its applications to equational matings. In Aït-Kaci, H. and Nivat, M., editors, *Resolution of Equations in Algebraic Structures*, volume 1, pages 151–216. Academic Press inc., New York.

[Gallier, 1986] Gallier, J. H. (1986). *Logic for Computer Science: Foundations of Automatic Theorem Proving*, volume 5 of *Computer Science and Technology Series*. Harper & Row, New York.

[Girard et al., 1989] Girard, J.-Y., Lafont, Y., and Taylor, P. (1989). *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press.

[Huet, 1972] Huet, G. (1972). *Constrained Resolution: A Complete Method for Type Theory*. PhD thesis, Case Western Reserve University.

[Huet, 1973] Huet, G. (1973). A mechanization of type theory. In *Proceeding of the third international joint conference on artificial intelligence*, pages 139–146.

[Huet, 1975] Huet, G. (1975). A unification algorithm for typed lambda calculus. *Theoretical Computer Science*, 1(1):27–57.

[Huet, 1976] Huet, G. (1976). *Résolution d'equations dans les langages d'ordre 1,2, ...,ω*. Thèse de Doctorat d'Etat, Université de Paris 7 (France).

[Hullot, 1980] Hullot, J.-M. (1980). Canonical forms and unification. In *Proceedings 5th International Conference on Automated Deduction, Les Arcs (France)*, pages 318–334.

[Jaffar and Lassez, 1987] Jaffar, J. and Lassez, J.-L. (1987). Constraint logic programming. In *Proceedings of the 14th Annual ACM Symposium on Principles Of Programming Languages, Munich (Germany)*, pages 111–119.

[Jouannaud and Kirchner, 1991] Jouannaud, J.-P. and Kirchner, C. (1991). Solving equations in abstract algebras: a rule-based survey of unification. In Lassez, J.-L. and Plotkin, G., editors, *Computational Logic. Essays in honor of Alan Robinson*, chapter 8, pages 257–321. The MIT press, Cambridge (MA, USA).

[Jouannaud and Kirchner, 1986] Jouannaud, J.-P. and Kirchner, H. (1986). Completion of a set of rules modulo a set of equations. *SIAM Journal of Computing*, 15(4):1155–1194. Preliminary version in Proceedings 11th ACM Symposium on Principles of Programming Languages, Salt Lake City (USA), 1984.

[Kirchner, 1985] Kirchner, C. (1985). *Méthodes et outils de conception systématique d'algorithmes d'unification dans les théories équationnelles*. Thèse de Doctorat d'Etat, Université Henri Poincaré – Nancy 1.

[Kirchner et al., 1990] Kirchner, C., Kirchner, H., and Rusinowitch, M. (1990). Deduction with symbolic constraints. *Revue d'Intelligence Artificielle*, 4(3):9–52. Special issue on Automatic Deduction.

[Kirchner and Ringeissen, 1997] Kirchner, C. and Ringeissen, C. (1997). Higher-Order Equational Unification via Explicit Substitutions. In *Proceedings 6th International Joint Conference ALP'97-HOA'97, Southampton (UK)*, volume 1298 of *Lecture Notes in Computer Science*, pages 61–75. Springer-Verlag.

[Kirchner, 1998] Kirchner, H. (1998). Orderings in Automated Theorem Proving. In Hoffman, F., editor, *Mathematical Aspects of Artificial Intelligence*, volume 55 of *Proceedings of Symposia in Applied Mathematics*, pages 55–95. American Mathematical Society.

[Klop et al., 1993] Klop, J., van Oostrom, V., and van Raamsdonk, F. (1993). Combinatory reduction systems: introduction and survey. *Theoretical Computer Science*, 121:279–308.

[Knuth and Bendix, 1970] Knuth, D. E. and Bendix, P. B. (1970). Simple word problems in universal algebras. In Leech, J., editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford.

[Lee and Plaisted, 1994] Lee, S.-J. and Plaisted, D. (1994). Use of replace rules in theorem proving. *Methods of Logic in Computer Science*, 1(2):217–240.

[Marché, 1994] Marché, C. (1994). Normalised rewriting and normalised completion. In Abramsky, S., editor, *Proceedings 9th IEEE Symposium on Logic in Computer Science, Paris (France)*, pages 394–403.

[McCune, 1997] McCune, W. (1997). Solution of the robbins problem. *Journal of Automated Reasoning*, 19(3):263–276.

[Nieuwenhuis and Rubio, 1994] Nieuwenhuis, R. and Rubio, A. (1994). AC-superposition with constraints: no AC-unifiers needed. In Bundy, A., editor, *Proceedings 12th International Conference on Automated Deduction, Nancy (France)*, volume 814 of *Lecture Notes in Artificial Intelligence*, pages 545–559. Springer-Verlag.

[Nutt et al., 1989] Nutt, W., Réty, P., and Smolka, G. (1989). Basic narrowing revisited. *Journal of Symbolic Computation*, 7(3 & 4):295–318. Special issue on unification. Part one.

[Peterson, 1983] Peterson, G. (1983). A technique for establishing completeness results in theorem proving with equality. *SIAM Journal of Computing*, 12(1):82–100.

[Peterson and Stickel, 1981] Peterson, G. and Stickel, M. E. (1981). Complete sets of reductions for some equational theories. *Journal of the ACM*, 28:233–264.

[Plaisted and Potter, 1991] Plaisted, D. A. and Potter, R. C. (1991). Term rewriting: Some experimental results. *Journal of Symbolic Computation*, 11(1&2):149–180.

[Plotkin, 1972] Plotkin, G. (1972). Building-in equational theories. *Machine Intelligence*, 7:73–90.

[Stickel, 1985] Stickel, M. (1985). Automated deduction by theory resolution. *Journal of Automated Reasoning*, 1(4):285–289.

[Stuber, 2001] Stuber, J. (2001). A Model-based Completeness Proof of Extended Narrowing And Resolution. In *1st Int. Joint Conf. on Automated Reasoning (IJCAR-2001), Siena, Italy*, volume 2083 of *LNCS*, pages 195–210. Springer.

[Terese (M. Bezem, J. W. Klop and R. de Vrijer, eds.), 2002] Terese (M. Bezem, J. W. Klop and R. de Vrijer, eds.) (2002). *Term Rewriting Systems.* Cambridge University Press.

[Vigneron, 1995] Vigneron, L. (1995). Positive Deduction modulo Regular Theories. In Büning, H. K., editor, *Annual Conference of the European Association for Computer Science Logic*, volume 1092 of *Lecture Notes in Computer Science*, pages 468–485, Paderborn (Germany). Springer-Verlag. Selected papers.

[Viry, 2002] Viry, P. (2002). Equational rules for rewriting logic. *Theoretical Computer Science*, 285(2):487–517.

Contents

Index

Unit´e de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unit´e de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unit´e de recherche INRIA Rhône-Alpes, 655, avenue de l'Europe, 38330 MONTBONNOT ST MARTIN
Unit´e de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unit´e de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
http://www.inria.fr
ISSN 0249-6399